

On the Exact and ϵ -Strong Simulation of (Jump) Diffusions

M. Pollock, A.M. Johansen, G.O. Roberts

Dept. of Statistics, University of Warwick, Coventry, CV4 7AL, UK

February 28, 2013

Abstract

This paper introduces a framework for simulating finite dimensional representations of (jump) diffusion sample paths over finite intervals, without discretisation error (*exactly*), in such a way that the sample path can be restored at any finite collection of time points. Within this framework we extend existing exact algorithms and introduce novel adaptive approaches. We consider an application of the methodology developed within this paper which allows the simulation of upper and lower bounding processes which almost surely constrain (jump) diffusion sample paths to any specified tolerance.

Keywords: Jump diffusions; Exact simulation; ϵ -strong simulation; Brownian path space probabilities

1 Introduction

A jump diffusion $V : \mathbb{R} \rightarrow \mathbb{R}$ is a Markov process. We consider jump diffusions defined as the solution to a stochastic differential equation (SDE) of the form (denoting $t- := \lim_{s \uparrow t} s$),

$$dV_t = \beta(V_{t-})dt + \sigma(V_{t-})dW_t + dJ_t^{\lambda, \nu} \quad V_0 = v \in \mathbb{R}, \quad t \in [0, T], \quad (1)$$

where $\beta : \mathbb{R} \rightarrow \mathbb{R}$ and $\sigma : \mathbb{R} \rightarrow \mathbb{R}_+$ denote the (instantaneous) drift and diffusion coefficients respectively, W_t is a standard Brownian Motion and $J_t^{\lambda, \nu}$ denotes a compound Poisson process. $J_t^{\lambda, \nu}$ is parameterised with (finite) jump intensity $\lambda : \mathbb{R} \rightarrow \mathbb{R}_+$, jump size coefficient $\nu : \mathbb{R} \rightarrow \mathbb{R}$ and as jumps distributed with density f_ν . All coefficients are themselves (typically) dependent on V_t . Regularity conditions are assumed to hold to ensure the existence of a unique non-explosive weak solution (see for instance [28, 30]). A discussion of conditions sufficient to allow the application of the methodology presented within this paper is given in Section 2.

Diffusions and jump diffusions are widely used across a number of application areas. An extensive literature exists in economics and finance, spanning from the Black-Scholes model [10, 25, 26] to the present [15, 4]. Other applications can be easily found within the physical [29] and life sciences [18, 19]. Motivated by such applications, there is considerable interest in finding solutions to SDEs. In particular, we would like to be able to draw sample paths from the measure induced by (1), denoted \mathbb{T}^ν . However, this is complicated as diffusion sample paths are infinite dimensional random variables (so at best we can hope to sample some finite dimensional subset of the sample path) and there doesn't typically exist an analytically tractable form for the transition density.

Diffusion sample paths can be simulated approximately at a finite collection of time points by *discretisation* [24, 30], noting that as Brownian motion has a Gaussian transition density then over short intervals the transition density of (1) can be approximated by one with fixed coefficients (by a continuity argument). This can be achieved by breaking the interval the sample path is to be simulated over into a fine mesh (for instance, of size Δt), then iteratively (at each mesh point) fixing the coefficients and then simulating the sample path to the next mesh point. For instance, in an *Euler* discretisation [21] of (1), the sample path is propagated between mesh points as follows,

$$V_{t+\Delta t} = \begin{cases} \beta(V_t)\Delta t + \sigma(V_t)\mathcal{N}(0, \Delta t) & \text{w.p. } \exp\{-\lambda(V_t)\Delta t\}, \\ \beta(V_t)\Delta t + \sigma(V_t)\mathcal{N}(0, \Delta t) + \nu(V_t) & \text{w.p. } 1 - \exp\{-\lambda(V_t)\Delta t\}. \end{cases} \quad (2)$$

It is hoped the simulated sample path (generated approximately at a finite collection of mesh points) can be used as a proxy for an entire sample path drawn exactly from \mathbb{T}^ν . More complex discretisation schemes exist (for

instance, by exploiting Itô's lemma to make higher order approximations or by local linearisation of the coefficients [24, 30]), but all suffer from common problems. In particular, minimising the approximation error (by increasing the mesh density) comes at the expense of increased computational cost, and further approximation or interpolation is needed to obtain the sample path at non-mesh points (which can be non-trivial). As a consequence, simulating sample paths by discretisation can be ill suited for subsequent application to a particular task.

Recently, a new class of *Exact Algorithms* for simulating sample paths at finite collections of time points without approximation error have been developed for both diffusions [9, 6, 7, 13] and jump diffusions [12, 17, 20]. These algorithms are based on rejection sampling, noting that sample paths can be drawn from the (target) measure \mathbb{T}^v by instead drawing sample paths from an equivalent proposal measure \mathbb{P}^v , and accepting or rejecting them with probability given by the Radon-Nikodým derivative of \mathbb{T}^v with respect to \mathbb{P}^v . However, as with discretisation schemes, given a simulated sample path at a finite collection of time points subsequent simulation of the sample path at any other intermediary point may require approximation or interpolation and may not be exact.

In this paper we introduce a novel mathematical framework for constructing exact algorithms which address this problem. In particular, instead of exactly simulating sample paths at finite collections of time points we instead focus on the extended notion of simulating *skeletons* which in addition characterise the entire sample path.

Definition 1.1 (Skeleton). *A skeleton (\mathcal{S}) is a finite dimensional representation of a diffusion sample path ($V \sim \mathbb{T}^v$), simulated without any approximation error by means of a proposal sample path drawn from an equivalent proposal measure (\mathbb{P}^v) and accepted with probability proportional to $\frac{d\mathbb{T}^v}{d\mathbb{P}^v}$, which is sufficient to restore the sample path at any finite collection of time points exactly with finite computation where $V(\omega)|\mathcal{S} \sim \mathbb{P}^v|\mathcal{S}$. A skeleton typically comprises information regarding the sample path at a finite collection of time points and path space information which ensures the sample path is almost surely constrained to some compact interval.*

Methodology for simulating skeletons (the size and structure of which is dependent on exogenous randomness) is driven by both computational and mathematical considerations (i.e. we need to ensure the required computation is finite and the skeleton is exact). Central to both notions is that the path space of the proposal measure \mathbb{P}^v can be broken into a discrete partition (a set of *layers*), and that the layer any particular sample path belongs to can be simulated.

Definition 1.2 (Layer). *A layer $R(V)$, is a function of a diffusion sample path $V \sim \mathbb{P}^v$ which determines the compact interval to which any particular sample path V is almost surely constrained.*

We show that, unlike existing exact algorithms, a valid exact algorithm can be constructed if it is possible to partition the proposal path space into layers, simulate unbiasedly which layer a proposal sample path belongs to and then, conditional on the layer, simulate a skeleton. Our exact algorithm framework for simulating skeletons is based on three principles for choosing a proposal measure and simulating a path space layer,

P1 - Principle 1 (Layer Construction). *The path space of the proposal measure \mathbb{P}^v , can be discretely partitioned and the layer a proposal sample path belongs to (i.e. $R(V) \sim \mathcal{R}$) can be unbiasedly simulated.*

P2 - Principle 2 (Proposal Exactness). *We can simulate from the probability law $\mathbb{P}^v \circ V^{-1} \circ R^{-1}$. In particular, conditional on $V_0 = v$, V_T and $R(V)$, we can simulate any finite collection of intermediate points of the trajectory of the proposal diffusion exactly.*

Together **P1** and **P2** ensure it is possible to simulate a skeleton. However, in addition we want to characterise the entire sample path and so we construct exact algorithms with the additional principle **P3**.

P3 - Principle 3 (Path Restoration). *Any finite collection of intermediate (inference) points, conditional on the skeleton, can be simulated exactly (i.e. we can sample $V_{t_1}(\omega), \dots, V_{t_n}(\omega) \sim \mathbb{P}^v|\mathcal{S}$)*

We also introduce a novel class of *Adaptive Exact Algorithms* for diffusions and jump diffusions, supported by new results for the simulation of Brownian path space probabilities (which are of separate interest) and *layered Brownian motion* (Brownian motion conditioned to remain in a layer). In addition, these results allow us to make methodological improvements to both the existing and new class of exact algorithm.

We apply the results developed in this paper to significantly extend ϵ -Strong Simulation methodology [8] (which allows the simulation of upper and lower bounding processes which almost surely constrain stochastic process

sample paths to any specified tolerance), from Brownian motion sample paths to a general class of jump diffusions, and introduce novel results to ensure the methodology in [8] can be implemented exactly. Finally, we provide a number of examples along with simulation results.

In summary, the main contributions of this paper are as follows:

- A mathematical framework for constructing exact algorithms and simulating diffusion and jump diffusion sample path skeletons (see Sections 3 and 4), under weaker conditions (see Sections 2 and 3).
- An extension of existing exact algorithms to satisfy **P3** (see Sections 3.1, 4.1, 4.2 and 6), including a number of methodological improvements (see Sections 3, 3.1, 4.1, 4.3, 4.4 and 5.2).
- A new class of *adaptive* exact algorithms for diffusions and jump diffusions (see Sections 3.2, 4.3 and 7).
- Methodology for simulating unbiasedly events of probability corresponding to various Brownian path space probabilities (see Section 5).
- Methodology for the ϵ -strong simulation of Brownian motion sample paths to ensure it can be initialised and is exact (see Sections 7 and 8).
- An extension to ϵ -strong simulation methodology from Brownian motion sample paths to a general class of jump diffusions (see Section 8).

This paper is organised as follows: In Section 2 we detail conditions sufficient to establish results necessary for applying the methodology in this paper. In Sections 3 and 4 we outline our exact algorithm framework for diffusions and jump diffusions respectively, including the adaptive exact algorithms. We extend existing layered Brownian bridge constructions in Section 6, introducing novel constructions for the adaptive exact algorithms in Section 7 (both of which rely on novel Brownian path space simulation results which are summarised in Section 5). Finally, in Section 8 we apply our methodology to the ϵ -strong simulation of (jump) diffusions.

2 Preliminaries & Conditions

The results underpinning this paper are for diffusions that are solutions to SDEs with unit volatility coefficient,

$$dX_t = \alpha(X_t) dt + dW_t, \quad X_0 = x \in \mathbb{R}, \quad t \in [0, T], \quad (3)$$

where W_t is Brownian Motion and $\alpha: \mathbb{R} \rightarrow \mathbb{R}$ is the drift function and satisfies the following (mild) conditions,

C1 - Condition 1. α has a linear growth bound, i.e. $\exists K > 0$ such that $|\alpha(x)| \leq K(1 + |x|)$.

C2 - Condition 2. α is continuously differentiable, i.e. $\alpha \in C^1$.

Applying the *Mean Value Theorem*, **C2** ensures α is locally Lipschitz and so (3) admits a unique weak solution [27]. Allowing \mathbb{Q}^x to denote the measure induced by (3) and \mathbb{W}^x as Wiener measure (on the interval $[0, T]$ with $X_0 = x$) and $A(u) := \int_0^u \alpha(y) dy$, we have,

R1 - As a result of **C1**, the Radon-Nikodým derivative of \mathbb{Q}^x with respect to \mathbb{W}^x exists and is given by Girsanov's formula (see for instance [5, 27]). Furthermore, as a consequence of **C2**, we have $A \in C^2$ and so we can apply *Itô's Lemma* to remove the stochastic integral (denoting $\phi(X_s) := \alpha^2(X_s)/2 + \alpha'(X_s)/2$),

$$\frac{d\mathbb{Q}^x}{d\mathbb{W}^x}(X) = \exp \left\{ \int_0^T \alpha(X_s) dX_s - \int_0^T \frac{1}{2} \alpha^2(X_s) ds \right\} = \exp \left\{ A(X_T) - A(x) - \int_0^T \phi(X_s) ds \right\}. \quad (4)$$

R2 - As a consequence of **C1** we have that A has a quadratic growth bound and so there exists some $T_0 < \infty$ such that $\forall T \leq T_0$:

$$c(y; x, T_0) := \int_{\mathbb{R}} \exp \left\{ A(y) - \frac{(y-x)^2}{2T} \right\} dy < \infty. \quad (5)$$

R3 - By **C2**, α and α' are bounded on compact sets. In particular, suppose $\exists \ell, v \in \mathbb{R}$ such that $\forall t \in [0, T]$, $X_t(\omega) \in [\ell, v] \implies L_X := L(X(\omega)) \in \mathbb{R}$, $U_X := U(X(\omega)) \in \mathbb{R}$ such that $\forall t \in [0, T]$, $\phi(X_t(\omega)) \in [L_X, U_X]$.

C1–C2 are sufficient to ensure that **R1–R3** hold (so we can implement the methodology in this paper), but can be relaxed if the Radon-Nikodým derivative of \mathbb{Q}^x with respect to a proposal measure exists and can be bounded (for instance see [16]).

Now, in order to tackle jump diffusions and heteroskedastic processes (as per (1)), further conditions are imposed,

C3 - Condition 3. $\beta \in C^1$.

C4 - Condition 4. $\sigma \in C^2$ and strictly positive.

C5 - Condition 5. λ is locally bounded.

Together with conditions **C1** and **C2**, **C3–C5** are sufficient for the *Lamperti transformation* for jump diffusions [24, 30] to be employed as follows (allowing methodology for the simulation of diffusions with unit volatility (3) to be extended to jump diffusions),

R4 - Let $\eta(V_t) =: X_t$ be a transformed process, where $\eta(V_t) := \int_{v^*}^{V_t} 1/\sigma(u) du$ (where v^* is an arbitrary element in the state space of V). Denoting by $N_t := \sum_{i \geq 1} \mathbb{1}_{t_i \leq t}$ a Poisson jump counting measure (with respect to \mathcal{F}_t^N) and applying Itô's formula for jump diffusions to find dX_t we have,

$$\begin{aligned} dX_t &= \left[\eta' dV_t + \eta'' (dV_t)^2 / 2 \right] + [\eta(V_{t-} + \nu(V_{t-})) - \eta(V_{t-})] dN_t \\ &= \underbrace{\left[\frac{\beta(\eta^{-1}(X_{t-}))}{\sigma(\eta^{-1}(X_{t-}))} - \frac{\sigma'(\eta^{-1}(X_{t-}))}{2} \right]}_{\alpha(X_{t-})} dt + dW_t + \underbrace{\left(\eta[\eta^{-1}(X_{t-}) + \nu(\eta^{-1}(X_{t-}))] - X_{t-} \right)}_{dJ_t^{\lambda, \mu}} dN_t. \end{aligned} \quad (6)$$

This transformation is typically possible for univariate diffusions and for many multivariate diffusions [1]. A significant class of multivariate diffusions can be simulated by direct application of our methodology, and ongoing work is aimed at extending these methodologies more broadly to multivariate diffusions (see [34]).

Denoting by \mathbb{Q}^x and \mathbb{W}^x , the measures induced by (6) and a driftless version of (6) respectively, allowing $\psi_1, \dots, \psi_{N_T}$ to denote the jump times in the interval $[0, T]$, $\psi_0 := 0$ and $\psi_{N_T+1-} := \psi_{N_T+1} := T$, we have that the following holds:

R5 - Under **C1–C2** the Radon-Nikodým derivative of \mathbb{Q}^x with respect to \mathbb{W}^x exists and is given by Girsanov's formula for jump diffusions [28, 24],

$$\frac{d\mathbb{Q}^x}{d\mathbb{W}^x}(X) = \exp \left\{ A(X_T) - A(x) - \int_0^T \phi(X_s) ds - \sum_{i=1}^{N_T} [A(X_{\psi_i}) - A(X_{\psi_i-})] \right\}. \quad (7)$$

3 Exact Algorithms

In this section we outline how to simulate skeletons for diffusion sample paths which can be represented (under the conditions in Section 2 and following the transformation in (6)), as the solution to SDEs with unit volatility,

$$dX_t = \alpha(X_t) dt + dW_t, \quad X_0 = x \in \mathbb{R}, \quad t \in [0, T]. \quad (8)$$

As discussed in Section 1, we can view exact algorithms as a class of rejection samplers operating on diffusion path space. In this section we begin by introducing rejection sampling and outline an (idealised) rejection sampler for simulating entire diffusion sample paths. However, for computational reasons, this idealised rejection sampler can't be implemented so we instead, with the aid of new results and algorithmic step reordering, address this issue and construct a rejection sampler for simulating sample path skeletons which only requires finite computation (a *retrospective rejection sampler*). We present two algorithmic interpretations of this retrospective rejection sampler. In Section 3.1 we present the *Bounded* and *Unbounded Exact Algorithms*, which are methodological extensions of existing exact algorithms [7, 17]. Finally, in Section 3.2 we introduce the novel *Adaptive Unbounded Exact Algorithm*.

Rejection sampling [33] is a standard Monte Carlo method in which we can sample from some (inaccessible) distribution π , provided there exists an accessible distribution g with respect to which π is absolutely continuous with bounded Radon-Nikodým derivative. In particular, if we can find a bound M such that $\sup_x \frac{d\pi}{dg} \leq M < \infty$, then drawing $X \sim g$ and accepting ($I = 1$) the draw with probability $P_g(X) := \frac{1}{M} \frac{d\pi}{dg}(X) \in [0, 1]$ then $(X|I = 1) \sim \pi$.

An (idealised) rejection sampler to draw entire sample paths from \mathbb{Q}^x (our target measure, induced by (8)) can be analogously constructed using proposal sample paths simulated from Wiener measure \mathbb{W}^x (which is the natural equivalent measure as (8) has unit volatility) which are accepted with probability $P_{\mathbb{W}^x}(X)$ (which is proportional to the Radon-Nikodým derivative of \mathbb{Q}^x with respect to \mathbb{W}^x as given in (4), noting that $A(x)$ is simply a constant).

Idealised Rejection Sampler

1 - Simulate $X \sim \mathbb{W}^x$

2 - With probability $P_{\mathbb{W}^x}(X) = \exp\left\{A(X_T) - \int_0^T \phi(X_s) ds\right\} / M$ accept, else reject and return to 1

Unfortunately, this idealised rejection sampler can't be implemented as it isn't possible to draw entire sample paths from \mathbb{W}^x (they're infinite dimensional random variables). Furthermore, $\phi(X_s)$ is typically unbounded and A only has a quadratic growth bound (see **R2**), so an appropriate bound ($M < \infty$) to evaluate the acceptance probability can't typically be found. Finally, it is not possible to evaluate the integral expression in the acceptance probability.

Clearly, as we can't simulate entire sample paths from \mathbb{Q}^x , the best we can hope for is to simulate a sample path skeleton. A retrospective rejection sampler (which only requires finite computation) can be constructed if it is possible to evaluate the acceptance probability of a proposal sample path without generating it in its entirety. If this were possible we could instead simulate a finite dimensional subset of the proposal sample path (a skeleton) then, given it is accepted, simulate any other aspect of the sample path required. The key to achieving this is noting that we don't need to evaluate the acceptance probability (which would require the entire sample path), but instead it is sufficient to find and simulate a binary random variable I , such that $\mathbb{P}(I = 1|\mathcal{S}) = P_{\mathbb{W}^x}(X)$.

We begin by finding a suitable bound ($M < \infty$) for the acceptance probability. To remove the unbounded function $A(X_T)$ from the acceptance probability we first simulate the end point from the law of *Biased Brownian Motion* (BBM) [9] and consider the resulting modification to the acceptance probability.

Definition 3.1 ([9, Prop. 3]). *Biased Brownian motion is the process $Z_t \stackrel{d}{=} (W_t|W_0 = x, W_T := y \sim h)$ with measure \mathbb{Z}^x , where $x, y \in \mathbb{R}$, $t \in [0, T]$ and h is defined as (by **R2** $\forall T \leq T_0 < \infty$, $h(y; x, T)$ is integrable),*

$$h(y; x, T) := \frac{1}{c(y; x, T)} \exp\left\{A(y) - \frac{(y - x)^2}{2T}\right\}, \quad (9)$$

Theorem 3.1 (Biased Brownian Motion [9, Prop. 3]). \mathbb{Q}^x is equivalent to \mathbb{W}^x with Radon-Nikodým derivative:

$$\frac{d\mathbb{Q}^x}{d\mathbb{W}^x}(X) \propto \exp\left\{-\int_0^T \phi(X_s) ds\right\}. \quad (10)$$

Sample paths can be drawn from \mathbb{Z}^x in two steps by first simulating the end point $X_T := y \sim h$ (although h doesn't have a tractable form a rejection sampler with Gaussian proposal can typically be constructed), then simulating the remainder of the sample path in the interval $(0, T)$ from the law of a Brownian bridge, $(X|X_0 = x, X_T = y) \sim \mathbb{W}_T^{x,y}$. As a direct consequence of (10), the acceptance probability of sample paths proposed from \mathbb{Z}^x is as follows,

$$P_{\mathbb{Z}^x}(X) = \exp\left\{-\int_0^T \phi(X_s) ds\right\} / M. \quad (11)$$

Now, to determine whether a simulated proposal sample path is accepted we construct and simulate a binary random variable I , such that $\mathbb{P}(I = 1|\mathcal{S}) = P_{\mathbb{Z}^x}(X)$. This can be achieved by means of an unbiased estimator of $P_{\mathbb{Z}^x}(X)$ which is almost surely constrained to the interval $[0, 1]$. Recalling that $\phi(X_{[0,T]})$ is bounded on compact sets (as a consequence of **R3**) note that if (after simulating the end point from BBM) we partition the path space

of $\mathbb{Z}^x|X_T$ into disjoint layers and unbiasedly simulate which layer our proposal sample path belongs to (as in **P1**, denoting $R_X \sim \mathcal{R}$ as the simulated layer), then an upper and lower bound for $\phi(X_{[0,T]})$ can be found conditional on this layer ($U_X \in \mathbb{R}$ and $L_X \in \mathbb{R}$ respectively). Now, letting $\mathbb{K}_{R(X)}$ be the law of $\kappa \sim \text{Poi}((U_X - L_X)T)$, \mathbb{U}_κ the distribution of $(\xi_1, \dots, \xi_\kappa) \sim U[0, T]^\kappa$, and considering a Taylor series expansion of the exponential function in (11) we have,

$$P_{\mathbb{Z}^x}(X) = \mathbb{E}_{\mathcal{R}} [P_{\mathbb{Z}^x|R(X)}(X)] = \mathbb{E}_{\mathcal{R}} \left[e^{-L_X T} \sum_{j=0}^{\infty} \frac{\exp\{-(U_X - L_X)T\} [(U_X - L_X)T]^j}{j!} \left(\int_0^T \frac{U_X - \phi(X_s)}{(U_X - L_X)T} ds \right)^j / M \right] \quad (12)$$

$$= \mathbb{E}_{\mathcal{R}} \left[e^{-L_X T} \mathbb{E}_{\mathbb{K}_{R(X)}} \left[\left(\int_0^T \frac{U_X - \phi(X_s)}{(U_X - L_X)T} ds \right)^\kappa \right] / M \right] = \mathbb{E}_{\mathcal{R}} \left[e^{-L_X T} \mathbb{E}_{\mathbb{K}_{R(X)}} \mathbb{E}_{\mathbb{U}_\kappa} \left[\prod_{i=1}^{\kappa} \left(\frac{U_X - \phi(X_{\xi_i})}{U_X - L_X} \right) \right] / M \right]. \quad (13)$$

Now, as we only need to find a bound $M < \infty$ such that our unbiased estimator of $P_{\mathbb{Z}^x}(X)$ lies in $[0, 1]$, then we can choose $M := e^{-L_X T}$. The proposal sample path can be accepted or rejected with probability given by (13). Details on how to simulate layer information and simulate intermediary points conditional on layer information is presented in Sections 6 and 7. Note that in order to evaluate (13) we only need to simulate layer information and κ ($< \infty$) intermediary points, so we can now construct a retrospective rejection sampler (exact algorithm) requiring only finite computation. We present two distinct interpretations of this. In Section 3.1 we present the *Bounded* and *Unbounded Exact Algorithms* and in Section 3.2 we present the *Adaptive Unbounded Exact Algorithm*. It should be noted that as a consequence of (13) we don't require an explicit lower bound for $\phi(X_{[0,T]})$ (unlike existing exact algorithms [9, 6, 7, 12, 20]), instead we note that under the conditions in Section 2 there almost surely exists a layer dependent lower bound which suffices.

3.1 Bounded & Unbounded Exact Algorithms

The *Unbounded Exact Algorithm* presented in Algorithm 3.1 is a retrospective rejection sampler directly based on simulating a finite dimensional proposal skeleton as suggested by (13), accepting the proposal with probability given by (13) and afterwards infilling the sample path at any other desired finite collection of time points.

Algorithm 3.1 – Unbounded Exact Algorithm (UEA)

- 1 - Simulate skeleton end point $X_T := y \sim h$
 - 2 - Simulate layer information $R_X \sim \mathcal{R}$
 - 3 - Simulate skeleton points $(X_{\xi_1}, \dots, X_{\xi_\kappa}) | R_X$
 - 3.1 - Simulate $\kappa \sim \text{Poi}((U_X - L_X)T)$ and skeleton times $\xi_1, \dots, \xi_\kappa \sim U[0, T]$
 - 3.2 - Simulate sample path at skeleton times $X_{\xi_1}, \dots, X_{\xi_\kappa} \sim \mathbb{W}_T^{x,y} | R_X$
 - 4 - With probability $\prod_{i=1}^{\kappa} \left[(U_X - \phi(X_{\xi_i})) / (U_X - L_X) \right]$, accept entire path, else reject and return to 1
 - * - Simulate inference times $X_{t_1}, \dots, X_{t_n} \sim \left(\otimes_{i=1}^{\kappa+1} \mathbb{W}_{\xi_{i-1}, \xi_i}^{X_{\xi_{i-1}}, X_{\xi_i}} \right) | R_X$
-

A number of alternate methods for simulating unbiasedly layer information (Step 2), layered Brownian bridges (Step 3), and the sample path at further inference times after acceptance (Step *), are given in Section 6. The skeleton of the accepted sample path includes any information simulated for the purpose of evaluating the acceptance probability (any subsequent simulation must be consistent with the skeleton). As such, the skeleton is composed of the start and simulated end point ($X_{\xi_0} := x$ and $X_{\xi_{\kappa+1}} := y$), skeletal points $(X_{\xi_1}, \dots, X_{\xi_\kappa})$ and layer R_X ,

$$\mathcal{S}_{\text{UEA}}(X(\omega)) := \left\{ (X_{\xi_i}, X_{\xi_i})_{i=0}^{\kappa+1}, R_X \right\}. \quad (14)$$

The UEA extends the notion of an exact algorithm beyond simulating a sample path at a finite collections of time points without approximation, to the simulation of skeletons which characterise the entire sample path and can be used to restore the sample path at any finite collection of inference time points. Any valid combination of layer and

layered Brownian bridge can be inserted in Algorithm 3.1 to reconstruct any existing exact algorithm for diffusions [9, 6, 7, 13] (which satisfy **P1–P2**). However, unlike existing exact algorithms, the UEA in addition satisfies **P3**. Step * of Algorithm 3.1 can't be conducted in existing exact algorithms as the layer (R_X) imparts information across the entire interval. In Section 6 we show that Step * is possible (with some additional computation), by augmenting the skeleton with sub-interval layer information (denoting $R_X^{[a,b]}$ as the layer for the sub-interval $[a, b]$),

$$\mathcal{S}'_{\text{UEA}}(X(\omega)) := \left\{ \left(\xi_i, X_{\xi_i} \right)_{i=0}^{\kappa+1}, R_X, \left(R_X^{[\xi_{i-1}, \xi_i]} \right)_{i=1}^{\kappa+1} \right\} = \left\{ \left(\xi_i, X_{\xi_i} \right)_{i=0}^{\kappa+1}, \left(R_X^{[\xi_{i-1}, \xi_i]} \right)_{i=1}^{\kappa+1} \right\}. \quad (15)$$

The augmented skeleton allows the sample path to be decomposed into conditionally independent paths between each of the skeletal points and so the layer R_X no longer imparts information across the entire interval $[0, T]$. As such additional inference points required in Step * of Algorithm 3.1 can now be directly simulated,

$$X_{t_1}, \dots, X_{t_n} \sim \mathbb{W}_T^{x,y} | \mathcal{S}'_{\text{UEA}} = \otimes_{i=1}^{\kappa+1} \left(\mathbb{W}_{\xi_{i-1}, \xi_i}^{X_{\xi_{i-1}}, X_{\xi_i}} | R_X^{[\xi_{i-1}, \xi_i]} \right). \quad (16)$$

In the particular case in which $\phi(X_{[0,T]})$ is almost surely bounded there is no need to simulate layer information in Algorithm 3.1, the skeleton (17) can be simulated from the law of a Brownian bridge and given the skeleton we can directly simulate inference points (so we satisfy **P3** and no skeleton augmentation is required). This leads to the *Exact Algorithm 1* (EA1) originally proposed in [6], which we term the *Bounded Exact Algorithm* (BEA).

$$\mathcal{S}_{\text{BEA}}(X(\omega)) := \left\{ \left(\xi_i, X_{\xi_i} \right)_{i=0}^{\kappa+1} \right\}. \quad (17)$$

Diffusion sample paths can be simulated by successive simulation of sample paths of shorter length over the required interval by applying the Markov property, noting the Radon-Nikodým derivative in (4) decomposes as follows (for any $t \in [0, T]$),

$$\frac{d\mathbb{Q}^x}{d\mathbb{W}^x}(X) = \left[\exp \left\{ A(X_t) - A(x) - \int_0^t \phi(X_s) ds \right\} \right] \cdot \left[\exp \left\{ A(X_T) - A(X_t) - \int_t^T \phi(X_s) ds \right\} \right]. \quad (18)$$

A direct application of this decomposition is used in Section 4. As the computational cost of simulating a sample path doesn't scale linearly with interval length this decomposition is widely applied in practise.

3.2 Adaptive Unbounded Exact Algorithm

Within this section we outline a novel *Adaptive Unbounded Exact Algorithm* (AUEA). To motivate this we revisit (12,13) and consider the Unbounded Exact Algorithm after the simulation of layer information,

$$0 \leq \exp \{ - (U_X - L_X) T \} \leq \exp \left\{ - \int_0^T [\phi(X_s) - L_X] ds \right\} = P_{\mathbb{Z}^x | R(X)}(X) \leq 1. \quad (19)$$

Reinterpreting the estimator in (13), we are exploiting the fact that $P_{\mathbb{Z}^x | R(X)}(X)$ is precisely the probability a Poisson process of rate 1 on the graph $\mathcal{G}_A := \{(a, b) : a \in [0, T], b \in [0, \phi(X_{[0,T]}) - L_X]\}$ contains no points. As this is a difficult space in which to simulate a Poisson process, we are instead simulating a Poisson process of rate 1 on the larger graph $\mathcal{G}_P := \{(a, b) : a \in [0, T], b \in [0, U_X - L_X]\}$ (which is easier as $U_X - L_X$ is a constant) and then conducting Poisson thinning (accepting the entire sample path if there are no Poisson points in $\mathcal{G}_A \subseteq \mathcal{G}_P$).

As \mathcal{G}_P can be much larger than \mathcal{G}_A the resulting exact algorithm can be inefficient and computationally expensive. In this section we propose an *adaptive* scheme which exploits the simulation of intermediate skeletal points of the proposal sample path in Algorithm 3.1. In particular, note that each simulated skeletal point implicitly provides information regarding the layer the sample path is contained within in both the sub-interval before and after it. As such, by simulating each point separately we can use this information to construct a modified proposal $\mathcal{G}_P^M \subseteq \mathcal{G}_P$, composed of a Poisson process with piecewise constant intensity, for the simulation of the remaining points.

In Algorithm 3.1, the simulation of a proposal Poisson process of intensity Δ_X on the interval $[0, T]$ to determine the skeletal time points (ξ_1, \dots, ξ_k) , can alternatively be simulated by exploiting the *exponential waiting time* property between successive events [23]. In particular, denoting T_1, \dots, T_k as the time between each event ξ_1, \dots, ξ_k , then the timing of the events can be simulated by successive $\text{Exp}(\Delta_X)$ waiting times while $\sum_i T_i \leq T$.

The independence of the points of a Poisson process allows us to simulate them in any convenient order. In our case it is likely the sample path at points closer to the mid-point of the interval will contain more information about the layer the entire sample path is contained within. As such, there is an advantage in simulating these points first. If we begin at the interval mid-point ($T/2$), we can find the point closest to it by simulating $\text{Exp}(2\Delta_X)$ random variable τ (we are simulating the first point at *either* side of the mid-point). As such, the simulated point (denoted ξ) will be with equal probability at either $T/2 - \tau$ or $T/2 + \tau$. Considering this in the context of (19), upon simulating ξ we have simply broken the acceptance probability into sub-intervals, the realisation of the sample path at the simulated time (X_ξ) providing a binary unbiased estimate of the central sub-interval (where $u \sim U[0, 1]$),

$$P_{\mathbb{Z}^s | R(X), X_\xi}(X) = \exp \left\{ - \int_0^{T/2-\tau} [\phi(X_s) - L_X] ds - \int_{T/2+\tau}^T [\phi(X_s) - L_X] ds \right\} \mathbb{E} \left(\mathbb{1} \left[u \leq \frac{U_X - \phi(X_\xi)}{U_X - L_X} \right] \middle| X_\xi \right). \quad (20)$$

If the central sub-interval is rejected the entire sample path can be discarded. However, if it is accepted then the acceptance of the entire sample path is conditional on the acceptance of *both* the left and right hand sub-intervals in (20), each of which has the same structural form as we originally had in (19). As such, for each we can simply iterate the above process until we have exhausted the entire interval $[0, T]$.

As outlined above our approach is an algorithmic reinterpretation, but otherwise identical, to Algorithm 3.1. However, we now have the flexibility to exploit the simulated skeletal point X_ξ , to simulate new layer information for the remaining sub-intervals conditional on the existing layer R_X (as outlined in more detail in Section 7). In particular, if we evaluate the left hand sub-interval in (20), we can find new layer information (denoted $R_X^{[0, \xi]}$) which will contain tighter bound information regarding the sample path ($\ell_X \leq \ell_X^{[0, \xi]} \leq X_{[0, \xi]}(\omega) \leq \nu_X^{[0, \xi]} \leq \nu_X$) and so (as a consequence of **R3**) can be used to infer tighter bounds for $\phi(X_{[0, \xi]})$ (denoted $U_X^{[0, \xi]} (\leq U_X)$ and $L_X^{[0, \xi]} (\geq L_X)$),

$$P_{\mathbb{Z}^s | R_X^{[0, \xi]}, X_\xi}^{[0, \xi]}(X) = \exp \left\{ - \int_0^{T/2-\tau} [\phi(X_s) - L_X] ds \right\} = \exp \left\{ - (L_X^{[0, \xi]} - L_X) \left(\frac{T}{2} - \tau \right) \right\} \exp \left\{ - \int_0^{T/2-\tau} [\phi(X_s) - L_X^{[0, \xi]}] ds \right\}. \quad (21)$$

The left hand exponential in (21) is a constant and it is trivial immediately reject the entire path with the complement of this probability. The right hand exponential of (21) has the same form as (19) and so the same approach as outlined above can be employed, but over the shorter interval $[0, T/2 - \tau]$ and with the lower rate $\Delta_X^{[0, \xi]} (\leq \Delta_X)$. As a consequence, the expected number of intermediary points required in order to evaluate the acceptance probability in (19) is lower than the UEA in Algorithm 3.1.

This leads to the novel *Adaptive Unbounded Exact Algorithm* (AUEA) detailed in Algorithm 3.2. We outline how to simulate (unbiasedly) layer information (Step 2), intermediate skeletal points (Step 3.1.2) and new layer information (Step 3.1.4) in a variety of ways in Section 7. Our iterative scheme outputs a skeleton comprising skeletal points and layer information for the intervals between consecutive skeletal points. The AUEA with this skeleton has the distinct advantage that **P1–P3** are satisfied directly. In particular, any intermediate inference points required after the skeleton has been simulated can be simulated directly (by application of Step 3.1.2 and Step 3.1.4 in Algorithm 3.2), without any augmentation of the skeleton (as in Algorithm 3.1).

$$\mathcal{S}_{\text{AUEA}}(X(\omega)) := \left\{ \left(\xi_i, X_{\xi_i} \right)_{i=0}^{K+1}, \left(R_X^{[\xi_{i-1}, \xi_i]} \right)_{i=1}^{K+1} \right\} \quad (22)$$

In Algorithm 3.2 we introduce simplifying notation, motivated by its recursive nature in which (as shown in (20)) the acceptance probability is iteratively decomposed into intervals which have been estimated and are yet to be estimated. Π denotes the set comprising information required to evaluate the acceptance probability for each of the intervals still to be estimated, $\Pi := \{\Pi_i\}_{i=1}^{|\Pi|}$. Each Π_i contains information regarding the time interval it applies to, the sample path at known points at either side of this interval and the associated simulated layer information, which we denote $[s_{\Pi(i)}, t_{\Pi(i)}]$, $x_{\Pi(i)} := X_{s_{\Pi(i)}}^{\Pi(i)}$, $y_{\Pi(i)} := X_{t_{\Pi(i)}}^{\Pi(i)}$ and $R_X^{\Pi(i)}$ respectively (where $s_{\Pi(i)}^- \leq s_{\Pi(i)} < t_{\Pi(i)} \leq t_{\Pi(i)}^+$). As before, $R_X^{\Pi(i)}$ can be used to directly infer bounds for ϕ for this specific sample path over the interval $[s_{\Pi(i)}, t_{\Pi(i)}]$ (namely $L_X^{\Pi(i)}$, $U_X^{\Pi(i)}$ and $\Delta_X^{\Pi(i)}$). We further denote $m_{\Pi(i)} := (s_{\Pi(i)} + t_{\Pi(i)})/2$, $d_{\Pi(i)} := (t_{\Pi(i)} - s_{\Pi(i)})/2$ and $\Xi := \Pi_1$.

Algorithm 3.2 – Adaptive Unbounded Exact Algorithm (AUEA)

- 1 - Simulate skeleton end point $X_T := y \sim h$
 - 2 - Simulate initial layer information $R_X \sim \mathcal{R}$, setting $\Pi := \{\Xi\} := \{[0, T], X_0, X_T, R_X\}$ and $\kappa = 0$
 - 3 - While $|\Pi| \neq 0$,
 - 3.1 - Simulate $\tau \sim \text{Exp}(2\Delta_X^\Xi)$. If $\tau > d_\Xi$ then set $\Pi := \Pi \setminus \Xi$ else
 - 3.1.1 - Set $\kappa = \kappa + 1$ and with probability $1/2$ set $\xi'_\kappa = m_\Xi - \tau$ else $\xi'_\kappa = m_\Xi + \tau$
 - 3.1.2 - Simulate $X_{\xi'_\kappa} \sim \mathbb{W}_{s-(\Xi), t+(\Xi)}^{x(\Xi), y(\Xi)} \Big| R_X^\Xi$
 - 3.1.3 - With probability $(1 - [U_X^\Xi - \phi(X_{\xi'_\kappa})] / \Delta_X^\Xi)$ reject path and return to 1
 - 3.1.4 - Simulate new layer information $R_X^{[s-(\Xi), \xi'_\kappa]}$ and $R_X^{[\xi'_\kappa, t+(\Xi)]}$ conditional on R_X^Ξ
 - 3.1.5 - With probability $(1 - \exp\{-[L_X^{[s-(\Xi), \xi'_\kappa]} + L_X^{[\xi'_\kappa, t+(\Xi)]} - 2L_X^\Xi][d_\Xi - \tau]\})$ reject path and return to 1
 - 3.1.6 - Set $\Pi := \Pi \cup \{[s_\Xi, m_\Xi - \tau], X_{s-}^\Xi, X_{\xi'_\kappa}^\Xi, R_X^{[s-(\Xi), \xi'_\kappa]}\} \cup \{[m_\Xi + \tau, t_\Xi], X_{\xi'_\kappa}^\Xi, X_{t+}^\Xi, R_X^{[\xi'_\kappa, t+(\Xi)]}\} \setminus \Xi$
 - 4 - Define skeletal points ξ_1, \dots, ξ_κ in temporal order from the set $\{\xi'_1, \dots, \xi'_\kappa\}$
 - * - Simulate inference times $X_{t_1}, \dots, X_{t_n} \sim \left(\otimes_{i=1}^{\kappa+1} \mathbb{W}_{\xi_{i-1}, \xi_i}^{X_{\xi_{i-1}}, X_{\xi_i}} \Big| R_X^{[\xi_{i-1}, \xi_i]}\right)$
-

4 Jump Exact Algorithms

In this section we extend the methodology of Section 3, constructing exact algorithms for simulating skeletons of jump diffusion sample paths which can be represented as the solution to the following SDE,

$$dX_t = \alpha(X_t) dt + dW_t + dJ_t^{\lambda, \mu} \quad X_0 = x \in \mathbb{R}, \quad t \in [0, T]. \quad (23)$$

Denoting \mathbb{Q}'^x as the measure induced by (23), we can draw sample paths from \mathbb{Q}'^x , by instead drawing sample paths from an equivalent proposal measure \mathbb{W}'^x (a natural choice being that induced by the driftless SDE in (24)), and accepting them with probability proportional to the Radon-Nikodým derivative of \mathbb{Q}'^x with respect to \mathbb{W}'^x .

$$dX_t = dW_t + dJ_t^{\lambda, \mu} \quad X_0 = x \in \mathbb{R}; \quad t \in [0, T] \quad (24)$$

The resulting Radon-Nikodým derivative (7) differs from that for diffusions (4) with the inclusion of an additional term, so the methodology of Section 3 can't be directly applied. However, (7) can be re-expressed in a product form similar to (4,18) (with $\psi_1, \dots, \psi_{N_T}$ denoting the jump times in the interval $[0, T]$, with $\psi_0 := 0$ and $\psi_{N_T+1} := T$),

$$\frac{d\mathbb{Q}'^x}{d\mathbb{W}'^x}(X) = \prod_{i=1}^{N_T+1} \left[\exp \left\{ A(X_{\psi_i-}) - A(X_{\psi_{i-1}}) - \int_{\psi_{i-1}}^{\psi_i-} \phi(X_s) ds \right\} \right]. \quad (25)$$

This product form of the Radon-Nikodým derivative is the basis for constructing *Jump Exact Algorithms*. Recall that decomposing the Radon-Nikodým derivative for diffusions justified the simulation of sample paths by successive simulation of sample paths of shorter length over the required interval (see (18)). Similarly, jump diffusion sample paths can be simulated by simulating diffusion sample paths of shorter length between consecutive jumps.

In this section we present three novel Jump Exact Algorithms (JEAs). In contrast with existing algorithms [12, 17, 20]), we note that the Bounded, Unbounded and Adaptive Unbounded Exact Algorithms in Section 3 can all be incorporated (with an appropriate choice of layered Brownian bridge construction) within any of the JEAs we develop. In Section 4.1 we present the *Bounded Jump Exact Algorithm*, which is a reinterpretation and methodological extension of [12], addressing the case where there exists an explicit bound for the intensity of the jump process. In Section 4.1 we present the *Unbounded Jump Exact Algorithm* which is an extension to existing exact algorithms [17, 20] in which the jump intensity is only locally bounded. Finally, in Section 4.3 we introduce an entirely novel *Adaptive Unbounded Jump Exact Algorithm* based on the adaptive approach of Section 3.2.

4.1 Bounded Jump Exact Algorithm

The case where the jump diffusion we want to simulate (23) has an explicit jump intensity bound ($\sup_{u \in [0, T]} \lambda(X_u) \leq \Lambda < \infty$) is of specific interest as the proposal jump times can be simulated in advance. In particular, proposal jump times, $\Psi_1, \dots, \Psi_{N_T^\Lambda}$ can be simulated according to a Poisson process with the homogeneous bounding intensity Λ over the interval $[0, T]$. A simple Poisson thinning argument [23] can be used to accept proposal jump times with probability $\lambda(X_{\Psi_j})/\Lambda$. As noted in [12], this approach allows the construction of a highly efficient algorithmic interpretation of the decomposition in (25). The interval can be broken into segments corresponding precisely to the intervals between proposal jump times, then iteratively between successive times, an exact algorithm (as outlined in Section 3) can be used to simulate a diffusion sample path skeleton. The terminal point of each skeleton can be used to determine whether the proposal jump time is accepted (and if so a jump simulated).

The *Bounded Jump Exact Algorithm* (BJEA) we outline in Algorithm 4.1 is a modification of that originally proposed [12] (where we define $\Psi_0 := 0$ and $\Psi_{N_T^\Lambda+1} := T$). In particular, we simulate the proposal jump times iteratively (exploiting the exponential waiting time property of Poisson processes [23] as in Section 3.2), noting that the best proposal distribution may be different for each sub-interval. Furthermore, we note that any of the exact algorithms we introduce in Section 3 can be incorporated in the BJEa (and so the BJEa will satisfy at least **P1–P2**). In particular, the BJEa skeleton is a concatenation of exact algorithm skeletons for the intervals between each proposal jump time, so to satisfy **P3** and simulate the sample path at further inference times (Step *), we either augment the skeleton if the exact algorithm chosen is the UEA (as discussed in Sections 3.1 and 6), or, if the exact algorithm chosen is the AUEA then simulate them directly (as discussed in Sections 3.2 and 7),

$$\mathcal{S}_{\text{BJEA}}(X(\omega)) := \bigcup_{j=1}^{N_T^\Lambda+1} \mathcal{S}_{\text{EA}}^j \quad (26)$$

Algorithm 4.1 – Bounded Jump Exact Algorithm (BJEA)

- 1 - Set $j = 0$. While $\Psi_j < T$,
 - 1.1 - Simulate $\tau \sim \text{Exp}(\Lambda)$. Set $j = j + 1$ and $\Psi_j = \Psi_{j-1} + \tau$
 - 1.2 - Apply exact algorithm to the interval $[\Psi_{j-1}, (\Psi_j \wedge T))$, outputting skeleton $\mathcal{S}_{\text{EA}}^j$
 - 1.3 - If $\Psi_j > T$ then set $X_T = X_{T-}$ else
 - 1.3.1 - With probability $\lambda(X_{\Psi_j})/\Lambda$ set $X_{\Psi_j} := X_{\Psi_j-} + f_v(X_{\Psi_j-})$ else set $X_{\Psi_j} := X_{\Psi_j-}$
 - * - Simulate inference times $X_{t_1}, \dots, X_{t_n} \sim \otimes_{j=1}^{N_T^\Lambda+1} \left(\otimes_{i=1}^{K_j+1} \mathbb{W}_{\xi_{j,i-1}, \xi_{j,i}}^{X_{\xi_{j,i-1}}, X_{\xi_{j,i}}} | R_{X[\xi_{j,0}, \xi_{j,K_j+1}]}^{\text{EA}} \right)$
-

4.2 Unbounded Jump Exact Algorithm

Considering the construction of a JEA under the weaker condition **C5** (in which we assume only that the jump intensity in (23) is locally bounded), it is not possible to first simulate the jump times as in Section 4.1. However, (as in Section 3 and as noted in [17, 20]), it is possible to simulate a layer $R_X \sim \mathcal{R}$, and then infer a jump intensity bound ($\lambda \leq \Lambda_X < \infty$) conditional on this layer. As such we can construct a JEA in this case by simply incorporating the jump intensity bound simulation within the layer framework of the UEA and AUEA.

The *Unbounded Jump Exact Algorithm* (UJEA) in Algorithm 4.2 is a JEA construction based on the UEA and extended from [20]. The UJEA is necessarily more complicated than the BJEa as simulating a layer in the UEA requires first simulating an end point. As a result, although (conditional on a layer) there exists a Poisson process in which to simulate proposal jump times, we can no longer segment the interval into sub-intervals according to the length of time until the next jump. Instead we need to simulate a diffusion sample path skeleton over the entire interval (along with all proposal jump times) and then determine the time of the first accepted jump and simulate it (if any). If a jump is accepted another diffusion sample path has to be proposed from the last accepted jump time to the end of the interval. This process is then iterated until no further jumps are accepted. The resulting UJEA

satisfies **P1–P2**, however, as a consequence of the layer construction, the jump diffusion skeleton is composed of the *entirety* of each proposed diffusion sample path skeleton (we can't apply the strong Markov property to discard the sample path skeleton after an accepted jump). In particular, as the simulated layer covers the entire interval (the sub-interval before and after an accepted jump) it imparts information across the interval, so the skeletal points in the interval after an accepted jump contain information about the sample path before the accepted jump.

$$\mathcal{S}_{\text{UJEA}}^{\text{UEA}}(X(\omega)) := \bigcup_{j=0}^{N_T^{\Lambda}} \left\{ \left(\xi_i^j, X_{\xi_i^j} \right)_{i=0}^{\kappa_j+1}, \left(\Psi_1^j, X_{\Psi_1^j} \right)_{i=1}^{N_T^{\Lambda,j}}, R_{X[\psi_j, T]}^j \right\} \quad (27)$$

The UJEA doesn't satisfy **P3** unless the skeleton is augmented (as with the UEA outlined in Sections 3.1 and 6), however, this is computationally expensive and is not recommended in practice (hence we have omitted Step * from Algorithm 4.2). Alternatively we could use the AUEA within the UJEA to directly satisfy **P3**, however it is more efficient in this case to implement the *Adaptive Unbounded Jump Exact Algorithm* in Section 4.3.

Algorithm 4.2 – *Unbounded Jump Exact Algorithm (UJEA)*

- 1 - Set $j = 0$ and $\psi_j = 0$
 - 1.1 - Simulate skeleton end point $X_T := y \sim h(y; X_{\psi_j}, T - \psi_j)$
 - 1.2 - Simulate layer information $R_{X[\psi_j, T]}^j \sim \mathcal{R}$ and infer $\Lambda_{X[\psi_j, T]}^j$
 - 1.3 - Simulate proposal jump times $N_T^{\Lambda,j} \sim \text{Poi}(\Lambda_{X[\psi_j, T]}^j(T - \psi_j))$ and $\Psi_1^j, \dots, \Psi_{N_T^{\Lambda,j}}^j \sim U[\psi_j, T]$
 - 1.4 - Simulate skeleton points and diffusion at proposal jump times $(X_{\xi_1^j}, \dots, X_{\xi_{\kappa}^j}, X_{\Psi_1^j}, \dots, X_{\Psi_{N(\Lambda,j,T)}^j}) | R_{X[\psi_j, T]}^j$
 - 1.4.1 - Simulate $\kappa_j \sim \text{Poi}([U_{X[\psi_j, T]}^j - L_{X[\psi_j, T]}^j] \cdot (T - \psi_j))$ and skeleton times $\xi_1^j, \dots, \xi_{\kappa}^j \sim U[\psi_j, T]$
 - 1.4.2 - Simulate sample path at times $X_{\xi_1^j}, \dots, X_{\xi_{\kappa}^j}, X_{\Psi_1^j}, \dots, X_{\Psi_{N(\Lambda,j,T)}^j} \sim \mathbb{W}_{\psi_j, T}^{X,y} | R_{X[\psi_j, T]}^j$
 - 1.5 - With probability $(1 - \prod_{i=1}^{\kappa_j} [U_{X[\psi_j, T]}^j - \phi(X_{\xi_i^j})] / (U_{X[\psi_j, T]}^j - L_{X[\psi_j, T]}^j))$, reject and return to 1.1
 - 1.6 - For i in 1 to $N_T^{\Lambda,j}$
 - 1.6.1 - With probability $\lambda(X_{\Psi_i^j}) / \Lambda_{X[\psi_j, T]}^j$ set $X_{\Psi_i^j-} = X_{\Psi_i^j}$, $X_{\Psi_i^j} := X_{\Psi_i^j-} + f_v(X_{\Psi_i^j})$, $\psi_{j+1} := \Psi_i^j$, $j = j + 1$, and return to 1.1
-

4.3 Adaptive Unbounded Jump Exact Algorithm

The novel *Adaptive Unbounded Jump Exact Algorithm* (AUJEA) which we present in Algorithm 4.3 is based on the AUEA and a reinterpretation of the UJEA. Considering the UJEA, note that if we simulate diffusion sample path skeletons using the AUEA then, as the AUEA satisfies **P3** directly, we can simulate proposal jump times *afterwards*. As such, we only need to simulate the next proposal jump time (as opposed to all of the jump times), which (as argued in Section 3.2), provides further information about the sample path. In particular, the proposal jump time necessarily lies between two existing skeletal times, $\xi_- \leq \Psi \leq \xi_+$, so the layer information for that interval, $R_X^{[\xi_-, \xi_+]}$ can be updated with layer information for each sub-interval $R_X^{[\xi_-, \Psi]}$ and $R_X^{[\Psi, \xi_+]}$ (the mechanism is detailed in Section 6). Furthermore, upon accepting a proposal jump time Ψ , the sample path skeleton in the sub-interval after Ψ contains no information regarding the skeleton preceding Ψ (so it can be discarded). As such, the AUJEA satisfies **P1–P3** and the skeleton is composed of only the accepted segments of each AUEA skeleton,

$$\mathcal{S}_{\text{AUJEA}}(X(\omega)) := \bigcup_{j=1}^{N_T^{\Lambda}+1} \mathcal{S}_{\text{AUEA}}^{[\psi_{j-1}, \psi_j]}(X(\omega)). \quad (28)$$

Algorithm 4.3 – Adaptive Unbounded Jump Exact Algorithm (AUJEA)

- 1 - Set $j = 0$ and $\psi_j = 0$
 - 2 - Apply AUEA to interval $[\psi_j, T]$, outputting skeleton $\mathcal{S}_{\text{AUEA}}^{[\psi_j, T]}$
 - 3 - Set $k = 0$ and $\Psi_k^j = \psi_j$. While $\Psi_k^j < T$,
 - 3.1 - Infer $\Lambda_{X[\Psi_k^j, T]}^j$
 - 3.2 - Simulate $\tau \sim \text{Exp}\left(\Lambda_{X[\Psi_k^j, T]}^j\right)$. Set $k = k + 1$ and $\Psi_k^j = \Psi_{k-1}^j + \tau$
 - 3.3 - If $\Psi_k^j \leq T$
 - 3.3.1 - Simulate $X_{\Psi_k^j} \sim \mathcal{S}_{\text{AUEA}}^{[\psi_j, T]}$
 - 3.3.2 - Simulate $R_X^{[\xi_-, \Psi_k^j]}$ and $R_X^{[\Psi_k^j, \xi_+]}$ and set $\mathcal{S}_{\text{AUEA}}^{[\psi_j, T]} := \mathcal{S}_{\text{AUEA}}^{[\psi_j, T]} \cup \left\{ X_{\Psi_k^j}, R_X^{[\xi_-, \Psi_k^j]}, R_X^{[\Psi_k^j, \xi_+]} \right\} \setminus R_X^{[\xi_-, \xi_+]}$
 - 3.3.3 - With probability $\lambda(X_{\Psi_k^j})/\Lambda_{X[\Psi_{k-1}^j, T]}^j$ set $X_{\Psi_k^j} = X_{\Psi_{k-1}^j}$, $X_{\Psi_k^j} := X_{\Psi_{k-1}^j} + f_v(X_{\Psi_{k-1}^j})$, $\psi_{j+1} := \Psi_k^j$, retain $\mathcal{S}_{\text{AUEA}}^{[\psi_j, \psi_{j+1}]}$, discard $\mathcal{S}_{\text{AUEA}}^{[\psi_{j+1}, T]}$, set $j = j + 1$ and return to 2
 - 4 - Define skeletal points χ_1, \dots, χ_m in temporal order from the time points in $\mathcal{S}_{\text{AUEA}} := \bigcup_{i=1}^{j+1} \mathcal{S}_{\text{AUEA}}^{[\psi_{i-1}, \psi_i]}$
 - * - Simulate inference times $X_{t_1}, \dots, X_{t_n} \sim \otimes_{i=1}^{m+1} \mathbb{W}_{[\chi_{i-1}, \chi_i]}^{[X_{\chi_{i-1}}, X_{\chi_i}]} | R_X^{[\chi_{i-1}, \chi_i]}$
-

4.4 An Extension to the Unbounded & Adaptive Unbounded Jump Exact Algorithms

As we don't know the timing of the next jump in the UJEA and AUJEA, in contrast with the BJEA, we simulate diffusion sample paths which are longer than necessary (so computationally more expensive), then (wastefully) partially discard them. To avoid this problem we could break the interval into segments and iteratively simulate diffusion sample paths of shorter length over the interval (as in (18)), thereby minimising the length of discarded segments beyond an accepted jump. However, the computational cost of simulating a sample path does not scale linearly with the interval it has to be simulated over, so the optimal length to decompose the interval is unknown.

It is possible to extend the UJEA and AUJEA based on this decomposition and Poisson superposition [23]. In particular, if it is possible to find a *lower* bound for the jump intensity $\lambda \downarrow \in (0, \lambda)$, then we can consider the target jump process as being the superposition of two jump processes (one of homogeneous intensity $\lambda \downarrow$ and the other with inhomogeneous intensity $\lambda - \lambda \downarrow$). As such we can simulate the timing of an accepted jump in the jump diffusion sample path under the homogeneous jump intensity $\lambda \downarrow$ by means of a $\tau \sim \text{Exp}(\lambda \downarrow)$ random variable. If $\tau \in [0, T]$ then there is no need to simulate proposal diffusion skeletons over the entire interval $[0, T]$, instead we can simulate them over $[0, \tau]$. Furthermore, we can modify the bounding jump intensity in the UJEA and AUJEA for generating the proposal jump times in the proposal diffusion sample path skeletons from Λ_X to $\Lambda_X - \lambda \downarrow$.

5 Brownian Path Space Simulation

In this section we present key results which we use to construct layered Brownian bridges in Sections 6 and 7. In Section 5.1 we outline a number of established results pertaining to the simulation of a variety of aspects of Brownian bridge sample paths. In Section 5.2 we consider known results (along with some methodological improvements) for simulating events corresponding to the probability that Brownian and Bessel bridge sample paths are contained within particular intervals. Finally, in Section 5.3 we present novel work in which we consider simulating probabilities corresponding to a more complex Brownian path space partitioning. Central to Sections 5.2 and 5.3 are Theorem 5.1 and Corollaries 5.1 and 5.2, which together form the basis for simulating events of unknown probabilities which can be represented as alternating Cauchy sequences.

Theorem 5.1 (Series Sampling [14, Chapter 4.5]). *An event of (unknown) probability $p \in [0, 1]$, where there exists monotonically increasing and decreasing sequences, $(S_k^+ : k \in \mathbb{Z}_{\geq 0})$ and $(S_k^- : k \in \mathbb{Z}_{\geq 0})$ respectively, such that $\lim_{k \rightarrow \infty} S_k^+ \downarrow p$ and $\lim_{k \rightarrow \infty} S_k^- \uparrow p$, can be simulated unbiasedly. In particular, a binary random variable $P := \mathbb{1}(u \leq p)$ can be constructed and evaluated (where $u \sim U[0, 1]$), noting that as there almost surely exists a finite $K := \inf\{k : u \notin (S_k^-, S_k^+)\}$ we have $\mathbb{1}(u \leq p) = \mathbb{1}(u \leq S_K^-)$.*

Corollary 5.1 (Linear Transformation). *Probabilities which are linear transformations or ratios of a collection of probabilities, each of which have upper and lower convergent sequences can be simulated by extension of Theorem 5.1. In particular, suppose $f : \mathbb{R}_+^m \rightarrow \mathbb{R}_+ \in C^1$ such that $|df/du_i(u)| > 0 \forall 1 \leq i \leq m$ and $u \in \mathbb{R}_+^m$ and that the probability $p := f(p_1, \dots, p_m)$ then defining the sequences $(T_k^{i,-} : k \in \mathbb{Z}_{\geq 0})$ and $(T_k^{i,+} : k \in \mathbb{Z}_{\geq 0})$ as follows,*

$$T_k^{i,-} = \begin{cases} S_k^{i,-} & \text{if } df/du_i > 0 \\ S_k^{i,+} & \text{if } df/du_i < 0 \end{cases}, \quad T_k^{i,+} = \begin{cases} S_k^{i,+} & \text{if } df/du_i > 0 \\ S_{k+1}^{i,-} & \text{if } df/du_i < 0 \end{cases}. \quad (29)$$

we have that $S_k^- := f(T_k^{1,-}, \dots, T_k^{m,-})$ is monotonically increasing and converges to p from below and $S_k^+ := f(T_k^{1,+}, \dots, T_k^{m,+})$ is monotonically decreasing and converges to p from above.

Corollary 5.2 (Retrospective Inversion Sampling [7, Prop. 1]). *If p can be represented as the limit of an alternating Cauchy sequence $(S_k : k \in \mathbb{Z}_{\geq 0})$, then splitting the sequence into subsequences composed of the odd and even terms respectively, then each subsequence will converge to p , one of which will be monotonically decreasing and the other monotonically increasing, so events of probability p can be simulated by extension of Theorem 5.1.*

Algorithm 5.1 outlines how to simulate events of probability p (applying Corollary 5.2), where p can be represented by an alternating Cauchy sequence (the even terms converging from below and the odd terms from above),

$$0 = S_0 \leq S_2 < S_4 < S_6 < \dots < p < \dots < S_5 < S_3 < S_1 \leq 1. \quad (30)$$

Algorithm 5.1 – Retrospective Inversion Sampling

- 1 - Simulate $u \sim U[0, 1]$ and set $k = 1$
 - 2 - While $u \in (S_{2k}, S_{2k+1})$, $k = k + 1$
 - 3 - If $u \leq S_{2k}$ then accept else reject
-

5.1 Simulating Brownian Bridges and Related Processes

The density of a Brownian bridge sample path $W_{s,t}^{x,y}$, at an intermediate time $q \in (s, t)$ is Gaussian with mean $\bar{w} := x + (q - s)(y - x)/(t - s)$ and variance $\sigma_w^2 := (t - q)(q - s)/(t - s)$ (so can be simulated directly). The joint distribution of the minimum value reached by $W_{s,t}^{x,y}$, and the time at which it is attained (τ, \hat{m}) , is given by [22],

$$\mathbb{P}(\hat{m} \in dw, \tau \in dq | W_s = x, W_t = y) \propto \frac{(w - x)(w - y)}{\sqrt{(t - q)^3(q - s)^3}} \exp \left\{ -\frac{(w - x)^2}{2(q - s)} - \frac{(w - y)^2}{2(t - q)} \right\} dw dq. \quad (31)$$

Analogously the maximum $(\tau, W_\tau := \hat{m})$, can be considered by reflection. As shown in [6] we can draw from (31) as outlined in Algorithm 5.2 which extends [6] noting that it is possible to condition the minimum to occur within a particular interval (i.e. simulate $(\tau, \hat{m}) | (\hat{m} \in [a_1, a_2])$ where $a_1 < a_2 \leq x \wedge y$).

Conditional on a Brownian bridge sample path minimum (or maximum), the law of the remainder of the trajectory is that of a *Bessel bridge*, which can be simulated by means of a 3-dimensional Brownian bridge of unit length conditioned to start and end at zero as outlined in [3] and Algorithm 5.3 (maximum by reflection).

Algorithm 5.2 – *Brownian Bridge Simulation at its Minimum Point (constrained to the interval $[a_1, a_2]$ where $a_1 < a_2 \leq x \wedge y$ and conditional on $W_s = x$ and $W_t = y$ (denoting $\text{IGau}(\mu, \lambda)$ as the inverse Gaussian distribution with mean μ and shape parameter λ)*

- 1 - Simulate $u_1 \sim U[M(a_2), M(a_1)]$ where $M(a) := \exp\{-2[xy + ((x \wedge y) - a)^2 - ((x \wedge y) - a) \cdot (x + y)] / (t - s)\}$ and $u_2 \sim U[0, 1]$
 - 2 - Set $\hat{m} := x - [\sqrt{(y - x)^2 - 2(t - s) \log(u_1)} - (y - x)] / 2$
 - 3 - If $u_2 < \frac{x - \hat{m}}{x + y - 2\hat{m}}$ then $V \sim \text{IGau}\left(\frac{y - \hat{m}}{x - \hat{m}}, \frac{(y - \hat{m})^2}{t - s}\right)$ else $\frac{1}{V} \sim \text{IGau}\left(\frac{x - \hat{m}}{y - \hat{m}}, \frac{(x - \hat{m})^2}{t - s}\right)$
 - 4 - Set $\tau := \frac{t - s}{1 + V}$
-

Algorithm 5.3 – *(Minimum) Bessel Bridge Simulation (at time $q \in (s, t)$ given $W_s = x, W_t = y$ and $W_\tau = \hat{m}$).*

- 1 - If $q < \tau$ then $r = s$ else $r = t$. Simulate $b_1, b_2, b_3 \stackrel{iid}{\sim} N\left(0, \frac{|\tau - q| \cdot |q - r|}{(\tau - r)^2}\right)$
 - 2 - Set $W_q := \hat{m} + \sqrt{|\tau - r|} \cdot \sqrt{\left(\frac{(W_r - \hat{m}) \cdot |\tau - q|}{|\tau - r|^{3/2}} + b_1\right)^2 + b_2^2 + b_3^2}$
-

5.2 Simulating Elementary Brownian Path Space Probabilities

In this section we briefly outline results pertaining to the probability that a Brownian bridge sample path is contained within a particular layer [2, 31] (Theorem 5.2) and how to simulate events of this probability [7] (Corollary 5.3). Similarly in Theorems 5.3 and 5.4 we outline a result (first shown in [7]) in which we show that the probability that a Bessel bridge sample path is contained within a particular layer can be represented as an infinite series.

Of particular importance for what follows is Corollary 5.5, in which we establish that it is possible (without any further assumption regarding the layer) to simulate events with a probability corresponding to the probability that a Bessel bridge sample path is contained within a particular layer by application Corollary 5.2.

Theorem 5.2 ([31, Theorem 3]). *The probability that a Brownian bridge sample path $W \sim \mathbb{W}_{s,t}^{x,y}$, remains in the interval $[\ell, v]$ (i.e. $\forall u \in [s, t] W_u \in [\ell, v]$) can be represented as an infinite series,*

$$\gamma_{s,t}^{\ell,v}(x, y) := \mathbb{P}(W \in [\ell, v]) = 1 - \sum_{j=1}^{\infty} \left\{ \varsigma_{s,t}^{\ell,v}(j; x, y) - \varphi_{s,t}^{\ell,v}(j; x, y) \right\}, \quad (32)$$

where $\varsigma_{s,t}^{\ell,v}(j; x, y) := \bar{\varsigma}_{s,t}^{\ell,v}(j; x, y) + \bar{\varsigma}_{s,t}^{-\ell,-v}(j; -x, -y)$, $\varphi_{s,t}^{\ell,v}(j; x, y) := \bar{\varphi}_{s,t}^{\ell,v}(j; x, y) + \bar{\varphi}_{s,t}^{-\ell,-v}(j; -x, -y)$ and,

$$\bar{\varsigma}_{s,t}^{\ell,v}(j; x, y) := \exp\left\{-\frac{2}{t-s}(|v - \ell|j + (\ell \wedge v) - x) \cdot (|v - \ell|j + (\ell \wedge v) - y)\right\}, \quad (33)$$

$$\bar{\varphi}_{s,t}^{\ell,v}(j; x, y) := \exp\left\{-\frac{2j}{t-s}(|v - \ell|^2 j + |v - \ell|(x - y))\right\}. \quad (34)$$

Corollary 5.3 ([7, Prop. 2]). *$\gamma_{s,t}^{\ell,v}(x, y)$ is an alternating Cauchy sequence, so events of probability $\gamma_{s,t}^{\ell,v}(x, y)$ can be simulated by retrospective inversion sampling (Corollary 5.2 and Algorithm 5.1) using the following sequence,*

$$S_{2k}^\gamma := 1 - \sum_{j=1}^k \left\{ \varsigma_{s,t}^{\ell,v}(j; x, y) - \varphi_{s,t}^{\ell,v}(j; x, y) \right\}, \quad S_{2k+1}^\gamma := S_{2k}^\gamma - \varsigma_{s,t}^{\ell,v}(k+1; x, y). \quad (35)$$

As shown in [7], Theorem 5.2 and Corollary 5.3 can be extended to consider simulating events with a probability corresponding to the probability a Bessel bridge sample path is contained within a particular layer. As indicated in Definition 5.1 we have to consider two possible cases where either of the end points attain the sample path minimum (or maximum) or not.

Definition 5.1. We allow $\delta_{s,t}^{\hat{m},v}(x,y) := \mathbb{1}\{\hat{m} < (x \wedge y)\} \cdot \delta_{s,t}^{\hat{m},v}(1;x,y) + \mathbb{1}\{\hat{m} = (x \wedge y)\} \cdot \delta_{s,t}^{\hat{m},v}(2;x,y)$ to denote the probability that a Bessel bridge sample path $W \sim \mathbb{W}_{s,t}^{x,y}|\hat{m}$, (with minimum \hat{m}) remains in the interval $[\hat{m}, v]$ ($\forall u \in [s, t] W_u \in [\hat{m}, v]$) where $\delta_{s,t}^{\hat{m},v}(1;x,y)$ and $\delta_{s,t}^{\hat{m},v}(2;x,y)$ are defined in Theorems 5.3 and 5.4 respectively.

Remark 5.1. We can similarly consider the probability that a Bessel bridge sample path $W \sim \mathbb{W}_{s,t}^{x,y}|\hat{m}$ (with maximum \hat{m}) remains in the interval $[\ell, \hat{m}]$ ($\forall u \in [s, t] W_u \in [\ell, \hat{m}]$) by a simple reflection argument of the above.

We first consider the case where the neither end point attains the Bessel bridge minimum (or maximum).

Theorem 5.3 ([7, Prop. 3]). The probability that a Bessel bridge sample path $W \sim \mathbb{W}_{s,t}^{x,y}|\hat{m}$, (with minimum $\hat{m} < (x \wedge y)$) remains in the interval $[\hat{m}, v]$ ($\forall u \in [s, t] W_u \in [\hat{m}, v]$) can be represented as an infinite series,

$$\delta_{s,t}^{\hat{m},v}(1;x,y) := \mathbb{P}(W \in [\hat{m}, v] | W \geq \hat{m}, (x \wedge y) > \hat{m}) = \frac{\gamma_{s,t}^{\hat{m},v}(x,y)}{1 - \exp\{-2(x - \hat{m})(y - \hat{m})/t\}}. \quad (36)$$

Corollary 5.4 ([7, Prop. 3]). Events of probability $\delta_{s,t}^{\hat{m},v}(1;x,y)$ can be simulated by application of retrospective inversion sampling (as per Corollaries 5.1, 5.2 and Algorithm 5.1) using the following sequence,

$$S_k^{\delta,1} := \frac{S_k^\gamma}{1 - \exp\{-2(x - \hat{m})(y - \hat{m})/t\}}. \quad (37)$$

We now consider the case where the either of end points attains the Bessel bridge minimum (or maximum).

Theorem 5.4 ([7, Prop. 3]). The probability that a Bessel bridge sample path $W \sim \mathbb{W}_{s,t}^{x,y}|\hat{m}$, (with minimum $\hat{m} = x < y$) remains in the interval $[\hat{m}, v]$ ($\forall u \in [s, t] W_u \in [\hat{m}, v]$) can be represented as an infinite series,

$$\delta_{s,t}^{\hat{m},v}(2;x,y) := \mathbb{P}(W \in [\hat{m}, v] | W \geq \hat{m}) = 1 - \frac{1}{(y - \hat{m})} \sum_{j=1}^{\infty} \{\psi_{s,t}^{\hat{m},v}(j;y) - \chi_{s,t}^{\hat{m},v}(j;y)\}, \quad (38)$$

where we denote,

$$\psi_{s,t}^{\hat{m},v}(j;y) := (2|v - \hat{m}|j - (y - \hat{m})) \exp\left\{-\frac{2|v - \hat{m}|j}{t - s}(|v - \hat{m}|j - (y - \hat{m}))\right\}, \quad (39)$$

$$\chi_{s,t}^{\hat{m},v}(j;y) := (2|v - \hat{m}|j + (y - \hat{m})) \exp\left\{-\frac{2|v - \hat{m}|j}{t - s}(|v - \hat{m}|j + (y - \hat{m}))\right\}. \quad (40)$$

Remark 5.2 ([7, Prop. 3]). We can similarly consider the probability a Bessel bridge sample path $W \sim \mathbb{W}_{s,t}^{x,y}|\hat{m}$, (with minimum $\hat{m} = y < x$) remains in the interval $[\hat{m}, v]$ by a simple reflection argument of Theorem 5.4.

Corollary 5.5. After the inclusion of the first $\hat{k} := \sqrt{(t - s) + |v - \hat{m}|^2} / (2|v - \hat{m}|)$ terms, $\delta_{s,t}^{\hat{m},v}(2;x,y)$ is an alternating Cauchy sequence, so events of probability $\delta_{s,t}^{\hat{m},v}(2;x,y)$ can be simulated by retrospective inversion sampling (as per Corollary 5.2 and Algorithm 5.1) using the following sequence (where $k \in \mathbb{N}$ such that $k \geq \hat{k}$),

$$S_{2k}^{\delta,2} := 1 - \frac{1}{(y - \hat{m})} \sum_{j=1}^k \{\psi_{s,t}^{\hat{m},v}(j;y) - \chi_{s,t}^{\hat{m},v}(j;y)\}, \quad S_{2k+1}^{\delta,2} := S_{2k}^{\delta,2} - \frac{1}{y - \hat{m}} \psi_{s,t}^{\hat{m},v}(k+1;y). \quad (41)$$

Proof. As $(y - \hat{m}) \in (0, (v - \hat{m})]$ then $\forall j$ we have $\psi_j, \chi_j \geq 0$. As such it is sufficient to show that $\forall j \geq \hat{k}$ that $\psi_j \geq \chi_j \geq \psi_{j+1} \geq \chi_{j+1} \geq \dots$ which can be shown by first showing that $\forall j \psi_j \geq \chi_j$ and then $\forall j \chi_j \geq \psi_{j+1}$. Considering ψ_j/χ_j if $j \geq \hat{k}$ then this is minimised when $y = \hat{m}$ and $\psi_j/\chi_j > 1$. Similarly considering χ_j/ψ_{j+1} if $j \geq \hat{k}$ then this is minimised when $y = v$ where $\chi_j/\psi_{j+1} > 1$ \square

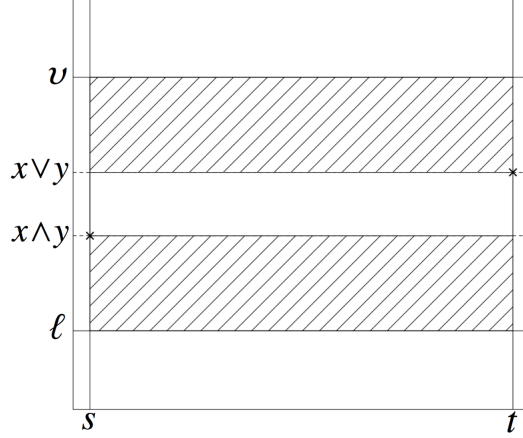


Figure 5.2: γ event – $W \in [\ell, v]$, $\hat{m} \in [\ell, (x \wedge y)]$, $\check{m} \in [(x \vee y), v]$

5.3 Simulating Brownian Path Space Probabilities

In this section we establish that the probability a Brownian bridge sample path, conditional on a number of intermediate points, has a minimum in a lower layer and a maximum in an upper layer (or in each sub-interval a minimum in a lower layer and a maximum in an upper layer) can be represented as an infinite series (Theorems 5.5 and 5.6 respectively), and events of this probability can be simulated (Corollaries 5.6 and 5.7 respectively).

In this section we introduce the following simplifying notation, $\hat{m}_{s,t} := \inf\{W_q; q \in [s, t]\}$, $\check{m}_{s,t} := \sup\{W_q; q \in [s, t]\}$, $\mathcal{Q} := \{q_1, \dots, q_n\}$, $\mathcal{W} := \{W_{q_1} = w_1, \dots, W_{q_n} = w_n\}$, $\mathcal{L} := \{\hat{m}_{s,q_1} \in [\ell_{s,q_1}^\downarrow, \ell_{s,q_1}^\uparrow], \dots, \hat{m}_{q_n,t} \in [\ell_{q_n,t}^\downarrow, \ell_{q_n,t}^\uparrow]\}$, $\mathcal{U} := \{\check{m}_{s,q_1} \in [v_{s,q_1}^\downarrow, v_{s,q_1}^\uparrow], \dots, \check{m}_{q_n,t} \in [v_{q_n,t}^\downarrow, v_{q_n,t}^\uparrow]\}$, $q_0 := s$ and $q_{n+1} := t$.

Theorem 5.5. *The probability a Brownian bridge sample path $W \sim \mathbb{W}_{s,t}^{x,y} | \mathcal{W}$ has a minimum $\hat{m}_{s,t} \in [\ell^\downarrow, \ell^\uparrow]$ and a maximum $\check{m}_{s,t} \in [v^\downarrow, v^\uparrow]$ can be represented as an infinite series,*

$$\begin{aligned} \rho_{s,t,x,y}^{(n), \ell^\downarrow, \ell^\uparrow, v^\downarrow, v^\uparrow}(\mathcal{Q}, \mathcal{W}) &:= \mathbb{P}(\hat{m}_{s,t} \in [\ell^\downarrow, \ell^\uparrow], \check{m}_{s,t} \in [v^\downarrow, v^\uparrow] | \mathcal{W}) \\ &= \left[\prod_{i=0}^n \gamma_{q_i, q_{i+1}}^{\ell^\downarrow, v^\uparrow} \right] - \left[\prod_{i=0}^n \gamma_{q_i, q_{i+1}}^{\ell^\uparrow, v^\uparrow} \right] - \left[\prod_{i=0}^n \gamma_{q_i, q_{i+1}}^{\ell^\downarrow, v^\downarrow} \right] + \left[\prod_{i=0}^n \gamma_{q_i, q_{i+1}}^{\ell^\uparrow, v^\downarrow} \right]. \end{aligned} \quad (42)$$

Proof. Follows by sample path inclusion-exclusion and the Markov property for diffusions,

$$\begin{aligned} \rho_{s,t,x,y}^{(n), \ell^\downarrow, \ell^\uparrow, v^\downarrow, v^\uparrow}(\mathcal{Q}, \mathcal{W}) &= \mathbb{P}(W \in [\ell^\downarrow, v^\uparrow] | \mathcal{W}) - \mathbb{P}(W \in [\ell^\uparrow, v^\uparrow] | \mathcal{W}) - \mathbb{P}(W \in [\ell^\downarrow, v^\downarrow] | \mathcal{W}) + \mathbb{P}(W \in [\ell^\uparrow, v^\downarrow] | \mathcal{W}) \\ &= \left[\prod_{i=0}^n \mathbb{P}(W_{[q_i, q_{i+1}]} \in [\ell^\downarrow, v^\uparrow] | W_{q_i}, W_{q_{i+1}}) \right] - \left[\prod_{i=0}^n \mathbb{P}(W_{[q_i, q_{i+1}]} \in [\ell^\uparrow, v^\uparrow] | W_{q_i}, W_{q_{i+1}}) \right] \\ &\quad - \left[\prod_{i=0}^n \mathbb{P}(W_{[q_i, q_{i+1}]} \in [\ell^\downarrow, v^\downarrow] | W_{q_i}, W_{q_{i+1}}) \right] + \left[\prod_{i=0}^n \mathbb{P}(W_{[q_i, q_{i+1}]} \in [\ell^\uparrow, v^\downarrow] | W_{q_i}, W_{q_{i+1}}) \right]. \quad \square \end{aligned}$$

Corollary 5.6. *Events of probability $\rho^{(n)}$ can be simulated by retrospective inversion sampling (as per Corollaries 5.1, 5.2 and Algorithm 5.1), noting that $\rho^{(n)}$ function of γ probabilities, using the following sequence,*

$$S_k^{\rho^{(n)}} := \left[\prod_{i=0}^n S_k^{\gamma(q_i, q_{i+1}, \downarrow, \uparrow)} \right] - \left[\prod_{i=0}^n S_{k+1}^{\gamma(q_i, q_{i+1}, \uparrow, \uparrow)} \right] - \left[\prod_{i=0}^n S_{k+1}^{\gamma(q_i, q_{i+1}, \downarrow, \downarrow)} \right] + \left[\prod_{i=0}^n S_k^{\gamma(q_i, q_{i+1}, \uparrow, \downarrow)} \right]. \quad (43)$$

Definition 5.2. *We define $\rho(s, q, t, x, w, y, \ell^\downarrow, \ell^\uparrow, v^\downarrow, v^\uparrow) := {}^{(1)}\rho_{s,t,x,y}^{\ell^\downarrow, \ell^\uparrow, v^\downarrow, v^\uparrow}(\mathcal{Q}, \mathcal{W})$, which is equal to ρ in [8].*

Theorem 5.6. *The probability a Brownian bridge sample path $W \sim \mathbb{W}_{s,t}^{x,y} | \mathcal{W}$, has in the sub-intervals induced between successive points in \mathcal{W} , a minimum and maximum in particular layers, \mathcal{L} and \mathcal{U} respectively, can be*

represented as an infinite series,

$$^{(n)}\beta_{s,t,x,y}^{\mathcal{L},\mathcal{U}}(\mathcal{Q},\mathcal{W}) := \mathbb{P}(\mathcal{L},\mathcal{U}|\mathcal{W}) = \prod_{i=0}^n \left[\gamma_{q_i,q_{i+1}}^{\ell\downarrow,v\uparrow} - \gamma_{q_i,q_{i+1}}^{\ell\uparrow,v\uparrow} - \gamma_{q_i,q_{i+1}}^{\ell\downarrow,v\downarrow} + \gamma_{q_i,q_{i+1}}^{\ell\uparrow,v\downarrow} \right]. \quad (44)$$

Proof. Follows by the Markov property for diffusions and sample path inclusion-exclusion,

$$\begin{aligned} ^{(n)}\beta_{s,t,x,y}^{\mathcal{L},\mathcal{U}}(\mathcal{Q},\mathcal{W}) &= \prod_{i=0}^n \left[\mathbb{P} \left(\hat{m}_{q_i,q_{i+1}} \in [\ell_{q_i,q_{i+1}}^\downarrow, \ell_{q_i,q_{i+1}}^\uparrow], \check{m}_{q_i,q_{i+1}} \in [v_{q_i,q_{i+1}}^\downarrow, v_{q_i,q_{i+1}}^\uparrow] \mid W_{q_i} = w_i, W_{q_{i+1}} = w_{i+1} \right) \right] \\ &= \prod_{i=0}^n \left[\mathbb{P} \left(W_{[q_i,q_{i+1}]} \in [\ell_{q_i,q_{i+1}}^\downarrow, v_{q_i,q_{i+1}}^\uparrow] \mid W_{q_i}, W_{q_{i+1}} \right) - \mathbb{P} \left(W_{[q_i,q_{i+1}]} \in [\ell_{q_i,q_{i+1}}^\uparrow, v_{q_i,q_{i+1}}^\uparrow] \mid W_{q_i}, W_{q_{i+1}} \right) \right. \\ &\quad \left. - \mathbb{P} \left(W_{[q_i,q_{i+1}]} \in [\ell_{q_i,q_{i+1}}^\downarrow, v_{q_i,q_{i+1}}^\downarrow] \mid W_{q_i}, W_{q_{i+1}} \right) + \mathbb{P} \left(W_{[q_i,q_{i+1}]} \in [\ell_{q_i,q_{i+1}}^\uparrow, v_{q_i,q_{i+1}}^\downarrow] \mid W_{q_i}, W_{q_{i+1}} \right) \right]. \quad \square \end{aligned}$$

Corollary 5.7. *Events of probability $\beta^{(n)}$ can be simulated by retrospective inversion sampling (as per Corollaries 5.1, 5.2 and Algorithm 5.1), noting that $\beta^{(n)}$ is a function of γ probabilities, using the following sequence,*

$$S_k^{\beta^{(n)}} := \prod_{i=0}^n \left[S_k^{\gamma(q_i,q_{i+1},\ell\downarrow,v\uparrow)} - S_{k+1}^{\gamma(q_i,q_{i+1},\ell\uparrow,v\uparrow)} - S_{k+1}^{\gamma(q_i,q_{i+1},\ell\downarrow,v\downarrow)} + S_k^{\gamma(q_i,q_{i+1},\ell\uparrow,v\downarrow)} \right]. \quad (45)$$

Definition 5.3. We define $\beta(s, t, x, y, \ell\downarrow, \ell\uparrow, v\downarrow, v\uparrow) := {}^{(0)}\beta_{s,t,x,y}^{\mathcal{L},\mathcal{U}}(\mathcal{Q},\mathcal{W})$, which is equal to β in [8].

6 Layered Brownian Bridge Constructions

In this section we outline how to construct and simulate finite dimensional skeletons of layered Brownian bridges for use within the UEA (Algorithm 3.1), which is in turn used within the BJE (Algorithm 4.1) and UJE (Algorithm 4.2). In particular, we address how to simulate unbiasedly layer information (Algorithm 3.1 Step 2), how to simulate intermediate skeletal points (Algorithm 3.1 Step 3) and how to simulate further inference times after acceptance of the proposed sample path (Algorithm 3.1 Step *).

We present two alternate layered Brownian bridge constructions based on extensions to existing exact algorithms. In Section 6.1 we present the *Bessel Approach*, which is a reinterpretation of part of the *Exact Algorithm 3* (EA3) proposed in [7], in which we incorporate the methodological improvements outlined in Sections 3 and 5 and introduce a novel approach for conducting Algorithm 3.1 Step * (which could not previously be achieved). As a consequence, the resulting (complete) UEA, with the inclusion of the Bessel approach, satisfies **P1–P3** (as opposed to only **P1–P2** in EA3 [7]). Finally, in Section 6.2 we outline a *Localised Approach* for constructing a layered Brownian bridge (based on the *Localised Exact Algorithm* (LEA) [13, 17]), showing that the resulting UEA only satisfies **P1–P2** and discussing the difficulties in conducting Algorithm 3.1 Step * and satisfying **P3**.

In both the Bessel and Localised approaches it isn't possible to directly simulate intermediate points conditional on a simulated layer (as required in Algorithm 3.1 Step 2). Instead, proposal sample path skeletons are simulated by intermediate Monte Carlo techniques, including Rejection Sampling (see Section 3) and demarginalisation [33].

Demarginalisation [33] is a technique whereby artificial extension of a density (with the incorporation of auxiliary variables) simplifies sampling from it. To illustrate this consider the case where we want to draw a sample $g(X)$, but this is not directly possible. However, suppose that with the introduction of an auxiliary variable Y , sampling from $g(X|Y)$ is possible and $g(X, Y)$ admits $g(X)$ as a marginal,

$$g(X) = \int_Y g(X|Y) g(Y) \, dY \quad (46)$$

We can sample from $g(X)$ by first sampling Y from $g(Y)$ and then sampling from $g(X|Y)$. This algorithm can be viewed as a black box to generate samples from $g(X)$ – Y can be simply marginalised out (i.e. ‘thrown’ away). Considering demarginalisation in the context of the exact algorithms, we can simulate any (auxiliary) aspect of the proposal diffusion sample path in addition to the skeleton to aid sampling. Provided the auxiliary information does not influence the acceptance probability then it is not part of the skeleton and doesn't need to be retained.

6.1 Bessel Approach (EA3)

The central idea in the *Bessel Approach* is that proposal Brownian bridge sample paths can be simulated jointly with information regarding the interval in which they are contained (Algorithm 3.1 Step 2), by means of a partitioning of Brownian bridge path space with an (arbitrary) increasing sequence, $\{a_i\}_{i \geq 0}$, $a_0 = 0$, which radiates outwards from the interval $[(x \wedge y), (x \vee y)]$ demarcating ‘layers’. For instance, the i^{th} layer is defined as follows,

$$\mathcal{I}_i = [(x \wedge y) - a_i, (x \vee y) + a_i]. \quad (47)$$

The (smallest) layer, $\mathcal{I} = \iota$, in which a particular Brownian bridge sample path is contained can be simulated by inversion and retrospective inversion sampling as in Algorithm 6.1. The CDF of \mathcal{I} can be written as follows (with reference to Theorem 5.2 and as shown in [7]),

$$\mathbb{P}(\mathcal{I} \leq \iota) = \mathbb{P}(W_{s,t}^{x,y} \in [(x \wedge y) - a_\iota, (x \vee y) + a_\iota]) = \gamma(s, t, x, y, (x \wedge y) - a_\iota, (x \vee y) + a_\iota). \quad (48)$$

Algorithm 6.1 – Simulation of a Brownian Bridge Layer

- 1 - Simulate $u \sim U[0, 1]$ and set $\iota = 1, k = 0$
 - 2 - While $u \in (S_{2k+1}^\gamma(s, t, x, y, (x \wedge y) - a_\iota, (x \vee y) + a_\iota), S_{2k}^\gamma(s, t, x, y, (x \wedge y) - a_\iota, (x \vee y) + a_\iota))$, $k = k + 1$
 - 3 - If $u \geq S_{2k}^\gamma$ set $\iota = \iota + 1$ and return to step 2 else set $\mathcal{I} = \iota$ and end
-

Now, in order to simulate intermediate points (Algorithm 3.1 Step 3), we need to condition on the layer simulated in Algorithm 6.1. In particular, denoting D_ι as the set of sample paths which are contained in the ι^{th} layer we have,

$$D_\iota = L_\iota \cup U_\iota \quad (49)$$

where,

$$L_\iota := \left\{ \hat{m}_{s,t} \in [(x \wedge y) - a_\iota, (x \wedge y) - a_{\iota-1}] \right\} \cap \left\{ \check{m}_{s,t} \in [(x \vee y), (x \vee y) + a_\iota] \right\}, \quad (50)$$

$$U_\iota := \left\{ \check{m}_{s,t} \in [(x \wedge y) - a_\iota, (x \wedge y)] \right\} \cap \left\{ \hat{m}_{s,t} \in [(x \vee y) + a_{\iota-1}, (x \vee y) + a_\iota] \right\}. \quad (51)$$

Directly simulating intermediate points from the law of D_ι , (denoted \mathbb{D}_ι) is not possible. Instead (as proposed in [7]) we can propose sample paths from the mixture measure $\mathbb{B}_\iota := \hat{\mathbb{M}}_\iota/2 + \check{\mathbb{M}}_\iota/2$ ($\hat{\mathbb{M}}_\iota$ and $\check{\mathbb{M}}_\iota$ being the law induced by \hat{M}_ι and \check{M}_ι respectively) and accept them with probability given by the Radon-Nikodým derivative of \mathbb{D}_ι with respect to \mathbb{B}_ι , where,

$$\hat{M}_\iota = \left\{ \hat{m}_{s,t} \in [(x \wedge y) - a_\iota, (x \wedge y) - a_{\iota-1}] \right\}, \quad \check{M}_\iota = \left\{ \check{m}_{s,t} \in [(x \vee y) + a_{\iota-1}, (x \vee y) + a_\iota] \right\}, \quad (52)$$

It was shown in [7] that \mathbb{D}_ι is absolutely continuous with respect to \mathbb{B}_ι with Radon-Nikodým derivative,

$$\frac{d\mathbb{D}_\iota}{d\mathbb{B}_\iota}(X) \propto \frac{\mathbb{1}(X \in D_\iota)}{1 + \mathbb{1}(X \in \hat{M}_\iota \cap \check{M}_\iota)}. \quad (53)$$

Sample paths can be drawn from \mathbb{D}_ι by proposing them from either $\hat{\mathbb{M}}_\iota$ or $\check{\mathbb{M}}_\iota$ with probability 1/2 and then accepting them with probability given by (53). For instance, conditional on simulating from $\hat{\mathbb{M}}_\iota$ we accept the sample path with probability 1 if the sample path maximum is contained within the $(\iota-1)^{\text{th}}$ layer or with probability 1/2 if it is contained between the $(\iota-1)^{\text{th}}$ and ι^{th} layer (and 0 otherwise). To aid simulation and ensure we simulate from $\hat{\mathbb{M}}_\iota$, we first simulate the sample path minimum $X_\tau = \hat{m}_{s,t}$ as per Algorithm 5.2, and subsequently simulate any required skeletal points ξ_1, \dots, ξ_k from a Bessel bridge as per Algorithm 5.3. As we can only simulate our sample path at a finite collection of points we can't directly evaluate (53). However, we can find probabilities for

these events and so simulate whether or not they occur by application of Corollaries 5.3 and 5.4 and Lemmata 5.4 and 5.5 (denoting $\chi_1, \dots, \chi_{\kappa+3}$ in temporal order from the set $\{\xi_1, \dots, \xi_\kappa, s, \tau, t\}$),

$$\mathbb{P}_{\hat{\mathbb{M}}_t}(X \in D_t) = \mathbb{P}\left(X \in [(x \wedge y) - a_t, (x \vee y) + a_t] \mid X_{\chi_1}, \dots, X_{\chi_{\kappa+3}}\right) = \prod_{i=1}^{\kappa+2} \delta_{\chi_i, \chi_{i+1}}^{\hat{m}_t(x \vee y) + a_t}(X_{\chi_i}, X_{\chi_{i+1}}), \quad (54)$$

$$\mathbb{P}_{\hat{\mathbb{M}}_t}(X \in \hat{M}_t \cap \check{M}_t) = \mathbb{P}_{\hat{\mathbb{M}}_t}(X \in D_t) - \prod_{i=1}^{\kappa+2} \delta_{\chi_i, \chi_{i+1}}^{\hat{m}_t(x \vee y) + a_{t-1}}(X_{\chi_i}, X_{\chi_{i+1}}). \quad (55)$$

As both (54) and (55) are probabilities which can be represented as a function of δ probabilities, then events of this probability can be simulated by retrospective inversion sampling (as per Corollaries 5.1, 5.2 and Algorithm 5.1). The synthesis of the above approach for simulating Brownian bridge conditional on the layer simulated in Algorithm 6.1 (i.e. conducting Algorithm 3.1 Step 3) leads to Algorithm 6.2.

Algorithm 6.2 – Layered Brownian Bridge Simulation (Bessel Approach)

- 1 - Simulate $u_1, u_2 \sim U[0, 1]$, set $j = k = 0$
 - 2 - Simulate Auxiliary Information (conditional on $I = \iota$)
 - 2.1 - If $u_1 \leq 1/2$ simulate minimum $X_\tau = \hat{m}_{s,t}$ ($= \ell_1 = \ell_2$) and set $v_1 = (x \vee y) + a_{t-1}$ and $v_2 = (x \vee y) + a_t$
 - 2.2 - If $u_1 > 1/2$ simulate maximum $X_\tau = \check{m}_{s,t}$ ($= v_1 = v_2$) and set $\ell_1 = (x \wedge y) - a_{t-1}$ and $\ell_2 = (x \wedge y) - a_{t-1}$
 - 3 - Simulate intermediate times $X_{\xi_1}, \dots, X_{\xi_k}$ from a Bessel Bridge conditional on X_τ
 - 4 - While $u_2 \in \left(\prod_{i=1}^{\kappa+2} S_{2j+1}^\delta(\ell_1, v_1), \prod_{i=1}^{\kappa+2} S_{2j}^\delta(\ell_1, v_1)\right)$, $j = j + 1$
 - 4.1 - If $u_2 \leq \prod_{i=1}^{\kappa+2} S_{2j+1}^\delta(\ell_1, v_1)$ then accept sample path
 - 4.2 - If $u_2 \geq \prod_{i=1}^{\kappa+2} S_{2j}^\delta(\ell_1, v_1)$ while $u_2 \in \left(\prod_{i=1}^{\kappa+2} S_{2k+1}^\delta(\ell_2, v_2), \prod_{i=1}^{\kappa+2} S_{2k}^\delta(\ell_2, v_2)\right)$, $k = k + 1$
 - 4.2.1 - If $u_2 \leq \prod_{i=1}^{\kappa+2} S_{2k+1}^\delta(\ell_2, v_2)$ then with probability 1/2 accept sample path, else return to step 1
 - 4.2.2 - If $u_2 \geq \prod_{i=1}^{\kappa+2} S_{2k}^\delta(\ell_2, v_2)$ then reject sample path and return to step 1
 - 5 - Discard or Retain Auxiliary Information
-

Upon accepting a proposed sample path skeleton within the UEA (as simulated by Algorithm 6.1 and Algorithm 6.2 and so satisfying **P1–P2**), we need to be able to simulate the sample path at further inference times (Algorithm 3.1 Step *) in order to satisfy **P3**. Any further simulation is conditional on information obtained constructing the sample path skeleton. In particular, our sample path belongs to D_t (by Algorithm 6.1), the sample path minimum (or maximum) belongs to a particular layer (as a consequence of the mixture proposal in (52)), we have simulated the sample path minimum (or maximum) (either $X_\tau = \hat{m}_{s,t}$ or $X_\tau = \check{m}_{s,t}$ by Algorithm 5.2) and skeletal points $(X_{\xi_1}, \dots, X_{\xi_k})$ and finally we have simulated whether the sample path maximum (or minimum) is contained in the first $(\iota - 1)$ layers or in the ι^{th} layer (by evaluating the Radon-Nikodým derivative in (53) by means of (54) and (55)). In summary, we have four possible sets of conditional information for our sample path X ,

$$\begin{aligned} S_1 &:= \{X_s, X_t, X \in D_t, \hat{m}_{s,t} \in [(x \wedge y) - a_t, (x \wedge y) - a_{t-1}], X_\tau = \hat{m}_{s,t}, X_{\xi_1}, \dots, X_{\xi_k}, \check{m}_{s,t} \in [(x \vee y), (x \vee y) + a_{t-1}]\} \\ S_2 &:= \{X_s, X_t, X \in D_t, \hat{m}_{s,t} \in [(x \wedge y) - a_t, (x \wedge y) - a_{t-1}], X_\tau = \hat{m}_{s,t}, X_{\xi_1}, \dots, X_{\xi_k}, \check{m}_{s,t} \in [(x \vee y) + a_{t-1}, (x \vee y) + a_t]\} \\ S_3 &:= \{X_s, X_t, X \in D_t, \check{m}_{s,t} \in [(x \vee y) + a_{t-1}, (x \vee y) + a_t], X_\tau = \check{m}_{s,t}, X_{\xi_1}, \dots, X_{\xi_k}, \hat{m}_{s,t} \in [(x \wedge y) - a_{t-1}, (x \wedge y)]\} \\ S_4 &:= \{X_s, X_t, X \in D_t, \check{m}_{s,t} \in [(x \vee y) + a_{t-1}, (x \vee y) + a_t], X_\tau = \check{m}_{s,t}, X_{\xi_1}, \dots, X_{\xi_k}, \hat{m}_{s,t} \in [(x \wedge y) - a_t, (x \wedge y) - a_{t-1}]\} \end{aligned}$$

The difficulty in simulating further intermediate inference times conditional on the above is that the layer information pertaining to the sample path minimum and maximum induces a dependency between the sub-interval in which we want to simulate an intermediate point, and all other sub-intervals. An additional complication arises

as we know precisely the minimum (or maximum) of the sample path, so the law we need to simulate further inference points from is that of a Bessel bridge conditioned to remain in a given interval.

However, the minimum (or maximum) simulated in Algorithm 6.2 Step 2 is auxiliary sample path information (as in (46)) and doesn't constitute part of the exact algorithm skeleton, so can be discarded. Furthermore, layer information regarding the sample path minimum and maximum is sufficient in determining a layer for the entire sample path. As such, reconsidering S_1 in light of the above (S_2, S_3, S_4 can be similarly considered) we have,

$$S_1 := \{X_s, X_t, X_{\xi_1}, \dots, X_{\xi_k}, \hat{m}_{s,t} \in [(x \wedge y) - a_t, (x \wedge y) - a_{t-1}], \check{m}_{s,t} \in [(x \vee y), (x \vee y) + a_{t-1}]\}$$

Now, to remove the layer induced dependency between sub-intervals we can unbiasedly simulate for each sub-interval a layer for the sample path minimum and maximum within that sub-interval, as outlined in Section 7.3 and Algorithm 7.3. Further intermediate inference points can then be simulated in a variety of ways as outlined in Sections 7.2, 7.5 and 7.6, Algorithm 7.5 and Algorithm 7.6.

6.2 Localised Approach

The *Localised Approach* is based on the layered Brownian bridge construction found in the *Localised Exact Algorithm* (LEA) originally proposed in [13, 17]. The LEA is an alternative exact algorithm based on the mathematical framework of EA3 (see [7]). However, we don't go into detail as to its construction as in the context of this paper it suffers from a number of computational weaknesses (in particular significant computation is required in order to satisfy **P3**) so is not well suited for the purposes of our paper. Instead, we briefly outline its construction and highlight which aspects of its construction present difficulties.

The key notion in the Localised approach is that rather than proposing sample path skeletons from \mathbb{Z}^x (where the end point $X_T := y \sim h$ is first simulated), the interval to be simulated ($[0, T]$) can be instead broken into a number of *bounded* segments (as in (18)). Each segment is successively simulated by means of simulating the first hitting time τ , of a Brownian motion proposal sample path (as outlined in [11]) of some user specified boundary symmetric around its start point (for instance, if $X_0 = x$ with boundary θ then $\tau := \inf\{s : X_s \notin [x - \theta, x + \theta]\}$), and simulating and accepting a sample path skeleton conditional on the simulated boundary (with a suitable modification of the acceptance probability to account for the modified proposal measure).

The benefit of the Localised approach is that simulating the first hitting time of a boundary acts as a layer for the bounded segment (i.e. $\forall u \in [0, \tau], X_u(\omega) \in [x - \theta, x + \theta]$) and so $\phi(X_{0,\tau})$ is conditionally bounded (as per **R3**) and furthermore a bound can be found for $A(X_\tau)$ in the acceptance probability. However, as with the UJEA and AUJEA this approach to simulating sample path skeletons can result in simulating skeletons for intervals exceeding that required (which is computationally wasteful), further complicated by the need to specify the boundary θ . Furthermore (as discussed in [20]), this methodology can't be used to simulate *conditioned* diffusion and jump diffusion sample path skeletons (sample paths conditioned to hit some specified end point). Finally, unlike the Bessel approach, the minimum or maximum that is simulated forms part of the skeleton and so can not be discarded. As such, the demarginalisation strategy taken in Section 6.1 in order to extend the UEA with the Bessel approach for simulating layered Brownian bridges to satisfy **P3** can't be conducted, so the skeleton can't be augmented.

7 Adaptive Layered Brownian Bridge Constructions

In Section 3.2 we proposed the AUEA (Algorithm 3.2) as an alternative to the UEA (Algorithm 3.1). In this section we outline how to simulate finite dimensional skeletons of layered Brownian bridges for use within the AUEA (and by extension the BJE (Algorithm 4.1) and AUJEA (Algorithm 4.3)). In particular, we present new results for simulating initial (intersection) layer information (Algorithm 3.2 Step 2 – Section 7.1), intermediate points conditional on the layer information (Algorithm 3.2 Step 3.1.2 – Section 7.2) and finally, new layer information for each sub-interval created by the intermediate point (Algorithm 3.2 Step 3.1.4 – Section 7.3).

We use the results we present in Sections 7.1–7.3 to outline two novel layered Brownian bridge constructions which can be used within the AUEA. In particular, in Sections 7.5 and 7.6 we introduce the *Intersection Layer Approach* and the *Hybrid Approach* respectively, both of which ensure the AUEA satisfies **P1–P3**.

7.1 Simulating an Initial Intersection Layer

Upon simulating a proposal Brownian bridge layer as per Algorithm 6.1 in Section 6.1, we know that our entire Brownian bridge sample path is contained within the ι^{th} layer, but is not contained within the $(\iota - 1)^{\text{th}}$ layer. Simulating sample path intermediary points is complicated by this conditional information (and as discussed in Section 6, it is not possible to simulate intermediary points directly). The novel approach we take in this paper is to simulate further layer information regarding the minimum and maximum of the proposed sample path (which together provide a sample path layer). To achieve this recall (with reference to Section 3 and (49,50,51)) that, having simulated a layer for our proposal Brownian bridge sample path as per Algorithm 6.1, we know the sample path belongs to the set of sample paths D_ι . We can then simply decompose the set D_ι into a disjoint union of possible sets our sample path belongs and simulate (unbiasedly) which our sample path belongs to,

$$D_\iota = L_\iota \cup U_\iota = \underbrace{(L_\iota \cap U_\iota)}_{D_{\iota,1}} \uplus \underbrace{(U_\iota^C \cap L_\iota)}_{D_{\iota,2}} \uplus \underbrace{(L_\iota^C \cap U_\iota)}_{D_{\iota,3}}, \quad (56)$$

This decomposition can be interpreted as the sample path attains the ι^{th} layer at both its minimum and maximum ($D_{\iota,1}$) or its minimum ($D_{\iota,2}$) or its maximum ($D_{\iota,3}$). We can simulate unbiasedly which set our sample path belongs to (which we call an *Intersection Layer*) by application of the following novel results and Algorithm 7.1.

Theorem 7.1 (Initial Intersection Layer). *The probability a Brownian bridge sample path is in $D_{\iota,1}$, given it is in D_ι , can be represented as follows (denoting $\ell_\downarrow := (x \wedge y) - a_\iota$, $\ell_\uparrow := (x \wedge y) - a_{\iota-1}$, $v_\downarrow := (x \vee y) + a_{\iota-1}$, $v_\uparrow := (x \vee y) + a_\iota$),*

$$\begin{aligned} p_{D_{\iota,1}} &:= \mathbb{P}(D_{\iota,1} \mid D_\iota, W_s = x, W_t = y) \\ &= \frac{\beta(s, t, x, y, \ell_\downarrow, \ell_\uparrow, v_\downarrow, v_\uparrow)}{\beta(s, t, x, y, \ell_\downarrow, \ell_\uparrow, v_\downarrow, v_\uparrow) + \beta(s, t, x, y, \ell_\downarrow, \ell_\uparrow, (x \vee y), v_\downarrow) + \beta(s, t, x, y, \ell_\uparrow, (x \wedge y), v_\downarrow, v_\uparrow)}. \end{aligned} \quad (57)$$

Proof. Follows by Bayes rule, Theorem 5.6 and the decomposition of D_ι in (56). \square

Corollary 7.1. *Events of probability $p_{D_{\iota,1}}$ can be simulated by retrospective inversion sampling (as per Corollaries 5.1, 5.2 and Algorithm 5.1), noting that $p_{D_{\iota,1}}$ is a function of β probabilities, using the following sequence,*

$$S_k^{D_{\iota,1}} := \frac{S_k^\beta(s, t, x, y, \ell_\downarrow, \ell_\uparrow, v_\downarrow, v_\uparrow)}{S_{k+1}^\beta(s, t, x, y, \ell_\downarrow, \ell_\uparrow, v_\downarrow, v_\uparrow) + S_{k+1}^\beta(s, t, x, y, \ell_\downarrow, \ell_\uparrow, (x \vee y), v_\downarrow) + S_{k+1}^\beta(s, t, x, y, \ell_\uparrow, (x \wedge y), v_\downarrow, v_\uparrow)}.$$

Remark 7.1. By symmetry we have that $\mathbb{P}(D_{\iota,2} \mid D_\iota, W_s = x, W_t = y) = \mathbb{P}(D_{\iota,3} \mid D_\iota, W_s = x, W_t = y)$.

Algorithm 7.1 – Simulation of an Initial Brownian Bridge Intersection Layer

- 1 - Simulate layer $\mathcal{I} = \iota$ as per Algorithm 6.1, simulate $u \sim U[0, 1]$ and set $j = 0$
 - 2 - While $u \in (S_{2j}^{D_{\iota,1}}, S_{2j+1}^{D_{\iota,1}})$, $j = j + 1$
 - 3 - If $u \leq S_{2j}^{D_{\iota,1}}$ then set $D_\iota = D_{\iota,1}$
 - 4 - If $u \geq S_{2j+1}^{D_{\iota,1}}$ then with probability 0.5 set $D_\iota = D_{\iota,2}$ else set $D_\iota = D_{\iota,3}$
-

7.2 Simulating Intersection Layer Intermediate Points

Having simulated an intersection layer we require a sampling scheme for simulating the conditional Brownian bridge at some intermediate time $q \in (s, t)$. As shown in [8], the density at q can be written as follows,

$$\pi(w) := \mathbb{P}(W_q = w \mid W_s, W_t, \hat{m}_{s,t} \in [\ell_{s,t}^\downarrow, \ell_{s,t}^\uparrow], \check{m}_{s,t} \in [v_{s,t}^\downarrow, v_{s,t}^\uparrow]) \propto \rho(s, q, t, x, w, y, \ell_{s,t}^\downarrow, \ell_{s,t}^\uparrow, v_{s,t}^\downarrow, v_{s,t}^\uparrow) \cdot N(\mu_w, \sigma_w^2). \quad (58)$$

A computationally efficient (though inexact) method of simulating from $\pi(w)$ was outlined in [8] based on inversion sampling and numerical methods. Here we show that it is possible to extend [8], and simulate from this

density exactly by means of composition sampling (see [32]) and rejection sampling. We will begin by considering the upper convergent bounding sequence of ρ ($\forall k \in \mathbb{Z}_{\geq 0}$ we have $\rho \leq S_{2k+1}^\rho$ and $\lim_{k \rightarrow \infty} S_{2k+1}^\rho = \rho$). Considering the decomposition of (43) into its elementary form in terms of ς and φ (see (33) and (34) respectively) note that it is composed of $K = 64(k+1)^2$ of these elementary terms.

Recalling that $\varsigma_{s,t}^{\ell,v}(j; x, y) := \bar{\varsigma}_{s,t}^{\ell,v}(j; x, y) + \bar{\varsigma}_{s,t}^{-\ell,-v}(j; -x, -y)$ and $\varphi_{s,t}^{\ell,v}(j; x, y) := \bar{\varphi}_{s,t}^{\ell,v}(j; x, y) + \bar{\varphi}_{s,t}^{-\ell,-v}(j; -x, -y)$ it can be shown that each of these terms has the structural form $\exp\{a_i + b_i w\}$. As such, we can find a bound for our target density (58) as follows ($c_i = \{0, 1\}$ denotes whether the density contribution is positive or negative),

$$\begin{aligned} \pi(w) \propto \rho(w) \cdot N(\mu_w, \sigma_w^2) &\leq S_{2k+1}^\rho \cdot N(\mu_w, \sigma_w^2) = \sum_{i=1}^K \left[(-1)^{c_i} \cdot \exp\{a_i + b_i w\} \cdot \mathbb{1}(w \in [\ell_i, v_i]) \cdot N(\mu_w, \sigma_w^2) \right] \\ &= \sum_{i=1}^K \left[(-1)^{c_i} \cdot \exp\{a_i + \mu_w b_i + b_i \sigma_w^2 / 2\} \cdot \mathbb{1}(w \in [\ell_i, v_i]) \cdot N(\mu_w + b_i \sigma_w^2, \sigma_w^2) \right] \\ &=: \sum_{i=1}^K \omega_i \cdot N(\mu_i, \sigma_w^2) \cdot \mathbb{1}(w \in [\ell_i, v_i]). \end{aligned} \quad (59)$$

Here we have a mixture of positively and negatively weighted truncated Normal densities (with common variance). Although each truncated Normal in the mixture is unique (due to the truncation points), a significant number have common mean. For instance, if $2k+1 = 1$ we have the Normal arising from $\bar{\varsigma}_{s,q}^{\ell,v}(j; x, w)$ and $\bar{\varsigma}_{s,q}^{\ell,v}(j; x, w)$ have the same mean. We exploit this by partitioning the interval that provides support for the target density (58) into sections corresponding to the truncation points. As a consequence, the resulting mixture density has a number of positive and negative elements which cancel each other out (i.e. they can be *netted* from one another). Defining $\omega^+ := (\omega \vee 0)$ we can find an upper bound by solely considering the mixture formed from the positive weights.

$$\begin{aligned} \pi(w) &\leq \sum_{i=1}^K \omega_i \cdot N(\mu_i, \sigma_w^2) \cdot \mathbb{1}(w \in [\ell_i, v_i]) \left[\mathbb{1}(w \in [\ell_{s,t}^\downarrow, \ell_{s,t}^\uparrow]) + \mathbb{1}(w \in [\ell_{s,t}^\uparrow, v_{s,t}^\downarrow]) + \mathbb{1}(w \in [v_{s,t}^\downarrow, v_{s,t}^\uparrow]) \right] \\ &\leq \sum_i N(\mu_i, \sigma_w^2) \mathbb{1}(w \in [\ell_i, v_i]) \left[\omega_{i,1}^+ \mathbb{1}(w \in [\ell_{s,t}^\downarrow, \ell_{s,t}^\uparrow]) + \omega_{i,2}^+ \mathbb{1}(w \in [\ell_{s,t}^\uparrow, v_{s,t}^\downarrow]) + \omega_{i,3}^+ \mathbb{1}(w \in [v_{s,t}^\downarrow, v_{s,t}^\uparrow]) \right] \\ &=: S_{2k+1}^{\rho,+} \cdot N(\mu_w, \sigma_w^2). \end{aligned} \quad (60)$$

By application of composition sampling (see [32]) we can simulate from $S_{2k+1}^{\rho,+} \cdot N(\mu_w, \sigma_w^2)$ by first choosing one of the truncated Normal densities partitioned on the interval $[L, \Upsilon]$ with probability proportional to,

$$\omega_{i,j}^+ \cdot \left[\Phi(\Upsilon | \mu_i, \sigma_w^2) - \Phi(L | \mu_i, \sigma_w^2) \right]. \quad (61)$$

As $w \sim S_{2k+1}^{\rho,+} \cdot N(\mu_w, \sigma_w^2)$ and we require $w \sim \rho \cdot N(\mu_w, \sigma_w^2)$ we accept this draw with probability,

$$P = \frac{\rho(w) \cdot N(w | \mu_w, \sigma_w^2)}{S_{2k+1}^{\rho,+}(w) \cdot N(w | \mu_w, \sigma_w^2)} = \frac{\rho(w)}{S_{2k+1}^{\rho,+}(w)}. \quad (62)$$

Events of probability P can be simulated by retrospective inversion sampling (as per Corollaries 5.1, 5.2 and Algorithm 5.1), noting that P is a function of ρ . The complete rejection sampler is presented in Algorithm 7.2.

Algorithm 7.2 – *Simulation of Intersection Layer Intermediate Points*

- 1 - Simulate $u \sim U[0, 1]$ and set $j = 1$
 - 2 - Simulate $w \sim S_1^{\rho,+} \cdot N(\mu_w, \sigma_w^2)$ for some $k \in \mathbb{Z}_{\geq 0}$
 - 2 - While $u \in \left(\frac{S_{2j}^\rho(w)}{S_1^{\rho,+}(w)}, \frac{S_{2j+1}^\rho(w)}{S_1^{\rho,+}(w)} \right)$, $j = j + 1$
 - 3 - If $u \leq \frac{S_{2j}^\rho(w)}{S_1^{\rho,+}(w)}$ then accept else reject
-

Simulating intermediate points as per Algorithm 7.2 is (typically) highly efficient as $S_1^{\rho,+} \cdot N(\mu_w, \sigma_w^2)$ (typically) forms a tight bound to $\pi(w)$ (as noted in [8]). If this is not the case, then sampling from the bounding density with $k > 1$ isn't usually effective as $S_{2k+1}^{\rho,+}$ is only formed by the positive netted components of S_{2k+1}^ρ . We present an alternative (exact) sampling scheme in Section 7.6 which can be used in this case based on the Bessel Approach.

Alternatively, it can be shown that as our target density $\pi(w) \leq S_{2k+1}^\rho \cdot N(\mu_w, \sigma_w^2)$ is constrained to the interval $[\ell_{s,t}^\downarrow, \ell_{s,t}^\uparrow]$, then by sampling the density $S_{2k+1}^\rho \cdot N(\mu_w, \sigma_w^2)$ on a sufficiently tight mesh over the interval $[\ell_{s,t}^\downarrow, \ell_{s,t}^\uparrow]$ and applying a convexity argument (noting the bounding density is a mixture of Normal densities), we can find a uniform bound for $S_{2k+1}^\rho \cdot N(\mu_w, \sigma_w^2)$ for any k (and so a uniform bound for $\pi(w)$). This uniform bound can be used to construct a rejection sampler in the case where $S_1^{\rho,+} \cdot N(\mu_w, \sigma_w^2)$ doesn't form a tight bound to $\pi(w)$ and we require a bounding density with $k > 1$. Furthermore, we conjecture that $\pi(w)$ is at most bimodal (with at most one mode in $[\ell_{s,t}^\uparrow, (\ell_{s,t}^\uparrow + v_{s,t}^\downarrow)/2]$ and at most one in $[(\ell_{s,t}^\uparrow + v_{s,t}^\downarrow)/2, v_{s,t}^\downarrow]$). Together these lead to highly efficient search methods for sampling exactly from $\pi(w)$.

7.3 Dissecting an Intersection Layer

Upon simulating intermediary points of a Brownian bridge sample path conditional on an intersection layer (for instance in Section 7.2), simulating further intermediary points in a sub-interval between any two existing consecutive points is more complicated as there is a dependency between all sub-intervals (induced by the intersection layer). To simplify this problem we can *dissect* an intersection layer into separate intersection layers for each pair of consecutive points by considering all possible dissections and unbiasedly simulating which one of these occurs.

To guide intuition we first consider the case where we have a single intermediary point ($W_q = w$) within an existing intersection layer ($W_s = x, W_t = y, \hat{m}_{s,t} \in [\ell_{s,t}^\downarrow, \ell_{s,t}^\uparrow], \check{m}_{s,t} \in [v_{s,t}^\downarrow, v_{s,t}^\uparrow]$) and we want to simulate separate intersection layers for the intervals $[s, q]$ and $[q, t]$ conditional on the known intersection layer and the simulated point. We begin by noting that the simulated point provides further detail on the interval in which the minimum and maximum lies. In particular, if $w \in [\ell_{s,t}^\downarrow, \ell_{s,t}^\uparrow]$ we have that $\hat{m}_{s,t} \in [\ell_{s,t}^\downarrow, w]$ and similarly if $w \in [v_{s,t}^\downarrow, v_{s,t}^\uparrow]$ then we have that $\check{m}_{s,t} \in [w, v_{s,t}^\uparrow]$. As such we denote $\ell_{s,t}^{\uparrow*} := (\ell_{s,t}^\uparrow \wedge w)$, $v_{s,t}^{\downarrow*} := (v_{s,t}^\downarrow \vee w)$ and we now have,

$$D_*(W_q = w) = \{\hat{m}_{s,t} \in [\ell_{s,t}^\downarrow, \ell_{s,t}^{\uparrow*}]\} \cap \{\check{m}_{s,t} \in [v_{s,t}^{\downarrow*}, v_{s,t}^\uparrow]\}. \quad (64)$$

The attainment of a particular layer in the interval $[s, t]$ by either the minimum or the maximum implies that the same layer is attained by the sample path in at least one of the sub-intervals $[s, q]$ or $[q, t]$. As such, in our case there are 9 possible (disjoint) bisections (which we denote as B_1 – B_9 where $B := D_*(W_q = w) = \bigoplus_{i=1}^9 B_i$) as illustrated in Figure 7.1. For instance, our sample path may lie in B_6 , which more formally has the form,

$$B_6 := \{ \{\hat{m}_{s,q} \in [\ell_{s,t}^{\uparrow*}, (x \wedge w)]\} \cap \{\check{m}_{s,q} \in [v_{s,t}^{\downarrow*}, v_{s,t}^\uparrow]\} \} \cup \{ \{\hat{m}_{q,t} \in [\ell_{s,t}^\downarrow, \ell_{s,t}^{\uparrow*}]\} \cap \{\check{m}_{q,t} \in [(w \vee y), v_{s,t}^{\downarrow*}]\} \}. \quad (65)$$

This notion can be extended to the case where we have multiple intermediate points ($\mathcal{W} := \{W_{q_1} = w_1, \dots, W_{q_n} =$

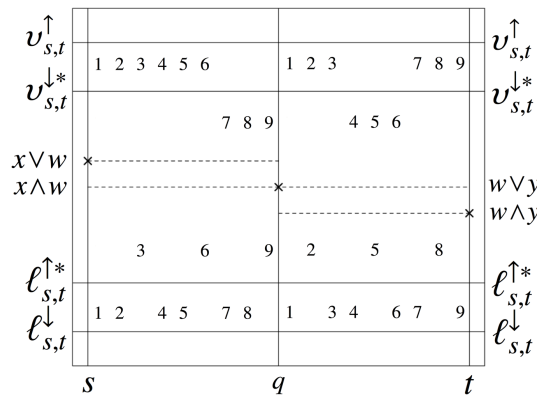


Figure 7.1: Illustration of 9 possible (disjoint) bisections

w_n)), and want to dissect the interval into separate intersection layers. In particular, we are dissecting a single

intersection layer into $(n + 1)$ intersection layers, each with a layer for the minimum and maximum in their own sub-interval. As the sample path minimum and maximum must exist in one of the intersection layers there are $b := (2^{(n+1)} - 1)^2$ possible dissections B_1^n, \dots, B_b^n . We can simulate unbiasedly which of these dissections our sample path lies in by application of the following novel results and Algorithm 7.3.

Theorem 7.2 (Intersection Layer Dissection). *The probability a Brownian bridge sample path is in B_i^n conditional on B and \mathcal{W} is as follows (denoting $\mathcal{L}(i)$ and $\mathcal{U}(i)$ as the lower and upper layer sets for B_i^n),*

$$p_{B_i^n} := \mathbb{P}\left(B_i^n \mid \hat{m}_{s,t} \in [\ell_{s,t}^\downarrow, \ell_{s,t}^\uparrow], \check{m}_{s,t} \in [v_{s,t}^\downarrow, v_{s,t}^\uparrow], W_s = x, W_t = y, \mathcal{W}\right) = \frac{{}^{(n)}\beta_{s,t,x,y}^{\mathcal{L}(i),\mathcal{U}(i)}(Q, \mathcal{W})}{{}^{(n)}\rho_{s,t,x,y}^{\ell_{s,t}^\downarrow, \ell_{s,t}^\uparrow, v_{s,t}^\downarrow, v_{s,t}^\uparrow}(Q, \mathcal{W})}. \quad (66)$$

Proof. Follows directly by Bayes rule, Theorem 5.5 and Theorem 5.6 \square

Remark 7.2 (Intersection Layer Bisection). *In the particular case where we have a single intermediary point then the probability a Brownian bridge sample path is in B_i (conditional on B and $W_q = w$) reduces to that in [8] (denoting $\ell_{s,q}^{\downarrow,i}, \ell_{s,q}^{\uparrow,i}, v_{s,q}^{\downarrow,i}, v_{s,q}^{\uparrow,i}$ and $\ell_{q,t}^{\downarrow,i}, \ell_{q,t}^{\uparrow,i}, v_{q,t}^{\downarrow,i}, v_{q,t}^{\uparrow,i}$ as the bounds for B_i in the interval $[s, q]$ and $[q, t]$ respectively),*

$$p_{B_i} = \frac{\beta(s, q, x, w, \ell_{s,q}^{\downarrow,i}, \ell_{s,q}^{\uparrow,i}, v_{s,q}^{\downarrow,i}, v_{s,q}^{\uparrow,i}) \cdot \beta(q, t, w, y, \ell_{q,t}^{\downarrow,i}, \ell_{q,t}^{\uparrow,i}, v_{q,t}^{\downarrow,i}, v_{q,t}^{\uparrow,i})}{\rho(s, q, t, x, w, y, \ell_{s,t}^\downarrow, \ell_{s,t}^{\uparrow*}, v_{s,t}^{\downarrow*}, v_{s,t}^\uparrow)}. \quad (67)$$

Corollary 7.2. *Events of probability p_{B_i} can be simulated by retrospective inversion sampling (as per Corollaries 5.1, 5.2 and Algorithm 5.1), noting that it is a function of ${}^{(n)}\beta_{s,t,x,y}^{\mathcal{L}(i),\mathcal{U}(i)}(Q, \mathcal{W})$ and ${}^{(n)}\rho_{s,t,x,y}^{\ell_{s,t}^\downarrow, \ell_{s,t}^\uparrow, v_{s,t}^\downarrow, v_{s,t}^\uparrow}(Q, \mathcal{W})$ probabilities, using the following sequence,*

$$S_k^{B(n,i)} := \frac{S_k^{\beta(n)}(s, t, x, y, Q, \mathcal{W}, \mathcal{L}(i), \mathcal{U}(i))}{S_{k+1}^{\rho(n)}(s, t, x, y, Q, \mathcal{W}, \ell_{s,t}^\downarrow, \ell_{s,t}^\uparrow, v_{s,t}^\downarrow, v_{s,t}^\uparrow)}. \quad (68)$$

Remark 7.3. *Unbiased simulation of the dissection the sample path lies in can be conducted by inversion sampling and a Cauchy sequence representation of the cdf of B (69). In particular, by application of Corollary 5.2, our sample path lies in B_j^n if for some $k > 0$ and $u \sim U[0, 1]$ we have $u \in (C_{2k+1}^{B(n,j-1)}, C_{2k}^{B(n,j)})$.*

$$C_k^{B(n,j)} := \sum_{i=1}^j S_k^{B(n,i)}. \quad (69)$$

Algorithm 7.3 – Dissecting an Intersection Layer

- 1 - Simulate $u \sim U[0, 1]$ and set $j = 1$ and $k = 0$
 - 2 - While $u \in (C_{2k}^{B(n,j)}, C_{2k+1}^{B(n,j)})$, $k = k + 1$
 - 3 - If $u \leq C_{2k}^{B(n,j)}$ set dissection layer $B = B_j$ else set $j = j + 1$ and return to step 2
-

7.4 Refining an Intersection Layer

Suppose we have already simulated layers for the maximum and minimum of our proposal Brownian bridge sample path ($\hat{m}_{s,t} \in [\ell_{s,t}^\downarrow, \ell_{s,t}^\uparrow]$ and $\check{m}_{s,t} \in [v_{s,t}^\downarrow, v_{s,t}^\uparrow]$), but we require more *refined* layer information (i.e. we want a set of narrower layers $|\ell_{s,t}^{\uparrow*} - \ell_{s,t}^{\downarrow*}| \leq |\ell_{s,t}^\uparrow - \ell_{s,t}^\downarrow|$ or $|v_{s,t}^{\uparrow*} - v_{s,t}^{\downarrow*}| \leq |v_{s,t}^\uparrow - v_{s,t}^\downarrow|$). This can be achieved by noting that the sample path falls in one of the following 4 possible (disjoint) intersection layer refinements (where $R := \uplus_{i=1}^4 R_i$) and simply unbiasedly simulating which one our sample path lies in (denoting $\ell_{s,t}^\uparrow \in [\ell_{s,t}^\downarrow, \ell_{s,t}^\uparrow]$ and $v_{s,t}^\uparrow \in [v_{s,t}^\downarrow, v_{s,t}^\uparrow]$),

$$\begin{aligned} R_1 &= \{\hat{m}_{s,t} \in [\ell_{s,t}^\downarrow, \ell_{s,t}^\uparrow]\} \cap \{\check{m}_{s,t} \in [v_{s,t}^\uparrow, v_{s,t}^\uparrow]\}, & R_2 &= \{\hat{m}_{s,t} \in [\ell_{s,t}^\uparrow, \ell_{s,t}^\uparrow]\} \cap \{\check{m}_{s,t} \in [v_{s,t}^\uparrow, v_{s,t}^\uparrow]\}, \\ R_3 &= \{\hat{m}_{s,t} \in [\ell_{s,t}^\downarrow, \ell_{s,t}^\uparrow]\} \cap \{\check{m}_{s,t} \in [v_{s,t}^\downarrow, v_{s,t}^\uparrow]\}, & R_4 &= \{\hat{m}_{s,t} \in [\ell_{s,t}^\uparrow, \ell_{s,t}^\uparrow]\} \cap \{\check{m}_{s,t} \in [v_{s,t}^\downarrow, v_{s,t}^\uparrow]\}. \end{aligned}$$

In a similar fashion to Section 7.3 we can simulate unbiasedly which of the intersection layer refinements our sample path lies in by application of the following established results and Algorithm 7.4.

Theorem 7.3 (Intersection Layer Refinement [8, Section 5.3]). *The probability a Brownian bridge sample path is in R_i conditional on R is as follows,*

$$p_{R_i} := \mathbb{P}\left(R_i \mid \hat{m}_{s,t} \in [\ell_{s,t}^\downarrow, \ell_{s,t}^\uparrow], \check{m}_{s,t} \in [v_{s,t}^\downarrow, v_{s,t}^\uparrow], W_s = x, W_t = y\right) = \frac{\beta(s, t, x, y, \ell_{s,t}^\downarrow, \ell_{s,t}^\uparrow, v_{s,t}^\downarrow, v_{s,t}^\uparrow)}{\beta(s, t, x, y, \ell_{s,t}^\downarrow, \ell_{s,t}^\uparrow, v_{s,t}^\downarrow, v_{s,t}^\uparrow)}. \quad (70)$$

Corollary 7.3 ([8, Section 5.3]). *Events of probability p_{R_i} can be simulated by retrospective inversion sampling (as per Corollaries 5.1, 5.2 and Algorithm 5.1), noting that it is a function of β probabilities, using the sequence,*

$$S_k^{R(i)} := \frac{S_k^\beta(s, t, x, y, \ell_{s,t}^\downarrow, \ell_{s,t}^\uparrow, v_{s,t}^\downarrow, v_{s,t}^\uparrow)}{S_{k+1}^\beta(s, t, x, y, \ell_{s,t}^\downarrow, \ell_{s,t}^\uparrow, v_{s,t}^\downarrow, v_{s,t}^\uparrow)}. \quad (71)$$

Remark 7.4. *By application of Corollary 5.2, unbiased simulation of the refinement the sample path lies in can be conducted by inversion sampling and a Cauchy sequence representation of the cdf of R (72). In particular, our sample path lies in R_j if for some $k > 0$ and $u \sim U[0, 1]$ we have $u \in (C_{2k+1}^{R(j-1)}, C_{2k}^{R(j)})$.*

$$C_k^{R(j)} := \sum_{i=1}^j S_k^{R(i)}. \quad (72)$$

Algorithm 7.4 – Refining an Intersection Layer

- 1 - Simulate $u \sim U[0, 1]$ and set $j = 1$ and $k = 0$
 - 2 - While $u \in (C_{2k}^{R(j)}, C_{2k+1}^{R(j)})$, $k = k + 1$
 - 3 - If $u \leq S_{2k}^{R(j)}$ set layer $R = R_j$ else set $j = j + 1$ and return to step 2
-

7.5 Intersection Layer Approach

The *Intersection Layer Approach* for constructing a layered Brownian bridge is a direct application of Sections 7.1–7.3. In particular, we simulate initial intersection layer information for the sample path (Algorithm 3.2 Step 2) by application of Algorithm 7.1. Now, in Algorithm 3.2 Step 3 we iteratively simulate skeletal (intermediate) points, then new intersection layer information conditional on the skeletal points. This can be achieved directly by application of Algorithm 7.2 and Algorithm 7.3 respectively.

We present the iterative AUEA Step 3 in Algorithm 7.5 which can be additionally used to conduct Step *. \mathcal{S} denotes the set containing all intersection layer information. The set is composed of $(n - 1)$ elements corresponding to the intervals between n existing time points. In particular, each element $(\mathcal{S}_{a,b})$ between two successive time points ($a < b$) contains information regarding the sample path at the end points and an upper and lower bound for both the minimum and maximum of the sample path in that interval $(\mathcal{S}_{a,b} := \{a, b, X_a, X_b, \ell_{a,b}^\downarrow, \ell_{a,b}^\uparrow, v_{a,b}^\downarrow, v_{a,b}^\uparrow\})$.

Algorithm 7.5 – Layered Brownian Bridge Simulation (Intersection Layer Approach)

- 1 - For each intermediate point required (q)
 - 1.1 - Select the appropriate existing intersection layer $\mathcal{S}_{a,b}$ from \mathcal{S} such that $q \in (a, b)$
 - 1.2 - Simulate X_q Algorithm 7.2
 - 1.3 - Bisect interval as per Algorithm 7.3 to find new intersection layers $\mathcal{S}_{a,q}$ and $\mathcal{S}_{q,b}$
 - 1.4 - Set $\mathcal{S} = \mathcal{S} \cup \{\mathcal{S}_{a,q}, \mathcal{S}_{q,b}\} \setminus \mathcal{S}_{a,b}$
-

7.6 Hybrid Approach

The *Hybrid Approach* for constructing a layered Brownian bridge is simply a modification of Algorithm 7.5 Step 2 (for reasons discussed in Section 7.2). An alternative is to simulate a single intermediate point using the Bessel approach outlined in Algorithm 6.2. However, a modification has to be made to the acceptance probability as the intersection layer provides more precise information regarding the interval in which both the minimum and maximum is contained. In particular, if we have simulated layer $D_{\iota,1}$ then with probability $1/2$ we propose the auxiliary minimum (else maximum) in the ι^{th} layer and then only accept the proposal sample path if the sample path maximum (else minimum) is contained between the $(\iota - 1)^{\text{th}}$ and ι^{th} layer. However, if we have simulated layer $D_{\iota,2}$ (else $D_{\iota,3}$) then we propose the auxiliary minimum (else maximum) in the ι^{th} layer and then only accept the proposal sample path if the sample path maximum (else minimum) is contained within the $(\iota - 1)^{\text{th}}$ layer.

8 ϵ -Strong Simulation of (Jump) Diffusions

In this section we outline a novel approach for simulating upper and lower bounding processes which almost surely constrain (jump) diffusion sample paths to any specified tolerance. We do this by means of a significant extension to the ϵ -Strong Simulation algorithm proposed in [8]. We consider applications of our extension, which are reliant on the new methodology presented elsewhere in this paper, in Sections 8.1 and 8.2 along with simulation results.

As originally proposed in [8] and presented in Algorithm 8.1, ϵ -Strong Simulation is an algorithm which simulates upper and lower convergent bounding processes (X^\uparrow and X^\downarrow) which enfold almost surely Brownian motion sample paths over some finite interval $[0, T]$. In particular, we have $\forall u \in [0, T]$ and some counter n ,

$$X_u^\downarrow(n+1) \leq X_u^\downarrow(n) \leq X_u \leq X_u^\uparrow(n+1) \leq X_u^\uparrow(n), \quad (73)$$

It should be noted that Algorithm 8.1 is an extension of that originally proposed in [8]. In particular, in contrast to [8], we can now simulate an initial intersection layer (Algorithm 8.1 Step 2) and simulate the intermediary points exactly (Algorithm 8.1 Step 3.1).

Algorithm 8.1 – ϵ -Strong Simulation of Brownian Motion sample paths (n bisections)

- 1 - Simulate $X_T := y \sim N(0, T)$ and set $i = 1$
 - 2 - Simulate initial intersection layer $\mathcal{S} := \mathcal{S}_{0,T} = \{0, T, X_0, X_T, \ell_{0,T}^\downarrow, \ell_{0,T}^\uparrow, v_{0,T}^\downarrow, v_{0,T}^\uparrow\}$ as per Algorithm 7.1
 - 3 - While $i \leq n$
 - 3.1 - For each of the 2^{i-1} intersection layers in \mathcal{S} (denoted $\mathcal{S}_{s,t}^j := \{s^j, t^j, X_s^j, X_t^j, \ell_{s,t}^{j,\downarrow}, \ell_{s,t}^{j,\uparrow}, v_{s,t}^{j,\downarrow}, v_{s,t}^{j,\uparrow}\}$)
 - 3.1.1 - Simulate X_q where $q := (s^j + t^j)/2$ conditional on $\mathcal{S}_{s,t}^j$ as per Algorithm 7.2
 - 3.1.2 - Bisect $\mathcal{S}_{s,t}^j$ into $\mathcal{S}_{s,q}^{j,1}$ and $\mathcal{S}_{q,t}^{j,2}$ as per Algorithm 7.3
 - 3.1.3 - For $\mathcal{S}_{s,q}^{j,1}$ and $\mathcal{S}_{q,t}^{j,2}$, while $|\ell_{s,t}^{j,*,\uparrow} - \ell_{s,t}^{j,*,\downarrow}| > \sqrt{(t^j - s^j)/4}$ or $|\nu_{s,t}^{j,*,\uparrow} - \nu_{s,t}^{j,*,\downarrow}| > \sqrt{(t^j - s^j)/4}$ then refine intersection layer as per Algorithm 7.4
 - 3.2 - Set $\mathcal{S} := \bigcup_{j=1}^{2^{i-1}} \{\mathcal{S}_{s,q}^{j,1} \cup \mathcal{S}_{q,t}^{j,2}\}$ and $i = i + 1$
-

The intersection layer information can be used to find the bounding processes in (73),

$$X_u^\uparrow(n) := \sum_{i=1}^{2^{n-1}} v_{s,t}^{i,\uparrow} \cdot \mathbb{1}(u \in [s^i, t^i]); \quad X_u^\downarrow(n) := \sum_{i=1}^{2^{n-1}} \ell_{s,t}^{i,\downarrow} \cdot \mathbb{1}(u \in [s^i, t^i]) \quad (74)$$

Furthermore we can find upper and lower bounds for the path integral,

$$X^\uparrow(n) := \sum_{i=1}^{2^{n-1}} v_{s,t}^{i,\uparrow} \cdot (t^i - s^i); \quad X^\downarrow(n) := \sum_{i=1}^{2^{n-1}} \ell_{s,t}^{i,\uparrow} \cdot (t^i - s^i) \quad (75)$$

It was shown in [8] (and summarised below) that the dominating processes converge almost surely in the supremum and L_1 norms, where the rate of convergence in L_1 is of the order $O(n^{-1/2})$.

Theorem 8.1 (Convergence [8, Prop. 3.1]). *Considering $X_u^\uparrow(n)$ and $X_u^\downarrow(n)$ as defined in (74), then convergence in the supremum norm holds in the limit as $n \rightarrow \infty$.*

$$\text{w.p. 1: } \lim_{n \rightarrow \infty} \sup_u |X_u^\uparrow(n) - X_u^\downarrow(n)| \rightarrow 0. \quad (76)$$

Theorem 8.2 (L_1 distance [8, Prop. 3.2]). *Considering the L_1 distance,*

$$|X^\uparrow - X^\downarrow|_1 = \int_0^T |X_u^\uparrow - X_u^\downarrow|_1 \, du \quad (77)$$

then,

$$2^{n/2} \times \mathbb{E} [|X^\uparrow - X^\downarrow|_1] = O(1) \quad (78)$$

Now, considering the ϵ -Strong Simulation of Jump Diffusions, note that upon simulating a jump diffusion sample path skeleton (as per the AUJEA), it has a form (see (22) and (28)) that can be used in Algorithm 8.1. As such, Algorithm 8.1 can be extended to jump diffusions (Algorithm 8.2), and Theorems 8.1 and 8.2 still hold.

Algorithm 8.2 – ϵ -Strong Simulation of Jump Diffusion sample paths (n bisections)

- 1 - Simulate jump diffusion skeleton as per Algorithm 3.2 to obtain initial intersection layer
 - 2 - Simulate further intersection layers as required (n bisections) as per Algorithm 8.1.
-

As far as we are aware there does not exist any other (exact or inexact) method for the ϵ -strong simulation of jump diffusions. The class of jump diffusions this methodology can be applied to is broad (the conditions outlined in Section 2 are sufficient) and motivate a number of avenues for future research and application. In particular, non-trivial characteristics of the diffusion path can be simulated (for instance extrema, hitting times, integrals) and can be applied to areas such as option pricing and the simulation of stochastic volatility models (which are currently being explored in related work). The precise implementation of Algorithm 8.1 and Algorithm 8.2 can be tailored to the specific application.

8.1 Application 1

In this section we consider the ϵ -strong simulation of jump diffusion sample paths which can be represented as solutions to the following SDE,

$$dX_t = -X_t \, dt + dW_t + dJ_t^{\lambda, \nu} \quad X_0 = 2, \quad t \in [0, 5], \quad \lambda(X_t) = \sin(X_t), \quad \nu(X_t) \sim f_\nu(X_t) = N(-X_t/2, 1) \quad (79)$$

In this case, as the jump intensity can be bounded, we can simulate sample path skeletons satisfying **P1–P3** using an AUEA incorporated within the BJEa. In particular, for this example we have that $\phi(X_t) := (X_t^2 - 1)/2$, $\phi(X_t) | (L_X, U_X) \in [-1/2, (\max\{(L_X)^2, (U_X)^2\} - 1)/2]$ and $\lambda(X_t) \leq 1 =: \Lambda$. Recall that in the BJEa the interval the sample path is to be simulated over $([0, 5])$, is broken into segments corresponding to proposed jump times (Ψ_1, \dots) . As such, considering the simulation of a diffusion sample path in the interval $[\Psi_1, \Psi_2]$ conditional on $X_{\Psi(1)}$ then the proposed end point is simulated according to BBM, $X_{\Psi(2)} \sim h(X_{\Psi(2)}; X_{\Psi(1)}, \Psi_2 - \Psi_1) \propto \exp\{-X_{\Psi(2)}^2/2 - (X_{\Psi(2)} - X_{\Psi(1)})^2 / [2(\Psi_2 - \Psi_1)]\}$.

In Figure 8.1.1 we consider a sample path skeleton simulated from the measure induced by (79) using the AUEA and BJEa. In Figure 8.1.2 we consider the ϵ -strong simulation of the simulated sample path as per Algorithm 8.2 with an increasing number of bisections. Finally, in Figure 8.1.3 we consider the ϵ -strong simulation of a number of other sample paths.

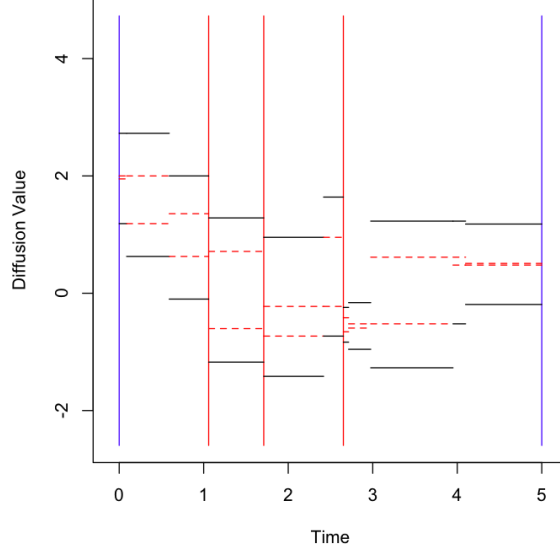


Figure 8.1.1: Sample path skeleton generated from the measure induced by (79). Vertical blue lines are the start and end point of the skeleton, vertical red lines are accepted jump times, horizontal black lines are almost sure bounds constraining the sample path and horizontal dotted red lines concern the layer attained by the sample path minimum and maximum in any given sub-interval.

8.2 Application 2

In this section we consider the ϵ -strong simulation of jump diffusion sample paths which can be represented as solutions to the following SDE,

$$dX_t = \sin(X_t) dt + dW_t + dJ_t^{1,\nu}, \quad X_0 = 0, t \in [0, 2], \lambda(X_t) = X_t^2, \nu(X_t) \sim f_\nu(X_t) = U[-X_t \wedge 0, -X_t \vee 0] \quad (80)$$

In this case the jump intensity is unbounded so to simulate sample path skeletons satisfying **P1–P3** we use the AU-JEA. In particular, for this example we have that $\phi(X_t) := (\sin^2(X_t) + \cos(X_t))/2 \in [-1/2, 5/8]$, $\lambda(X_t) | (L_X, U_X) \leq \max\{(L_X)^2, (U_X)^2\}$ and the end point simulated according to BBM, $X_T := y \sim h \propto (\exp\{-\cos(y) - y^2/6\})$.

In contrast with Section 8.1, in this section we instead consider the ϵ -strong simulation of jump diffusions generated from the measure induced by (79) to some specified tolerance ϵ (such that $X \uparrow - X \downarrow \leq \epsilon$). As such we appropriately modify Algorithm 8.1 and Algorithm 8.2 and instead bisect intervals in which the upper and lower bounds for the sample path are less sharp.

In Figure 8.2.1(a) we consider a sample path skeleton generated from the measure induced by (79) and in Figure 8.2.1(b) we consider the ϵ -strong simulation of this skeleton to the tolerance $\epsilon = 0.5$. In Figure 8.2.2 we consider the ϵ -strong simulation of a number of other sample paths to variety of tolerances.

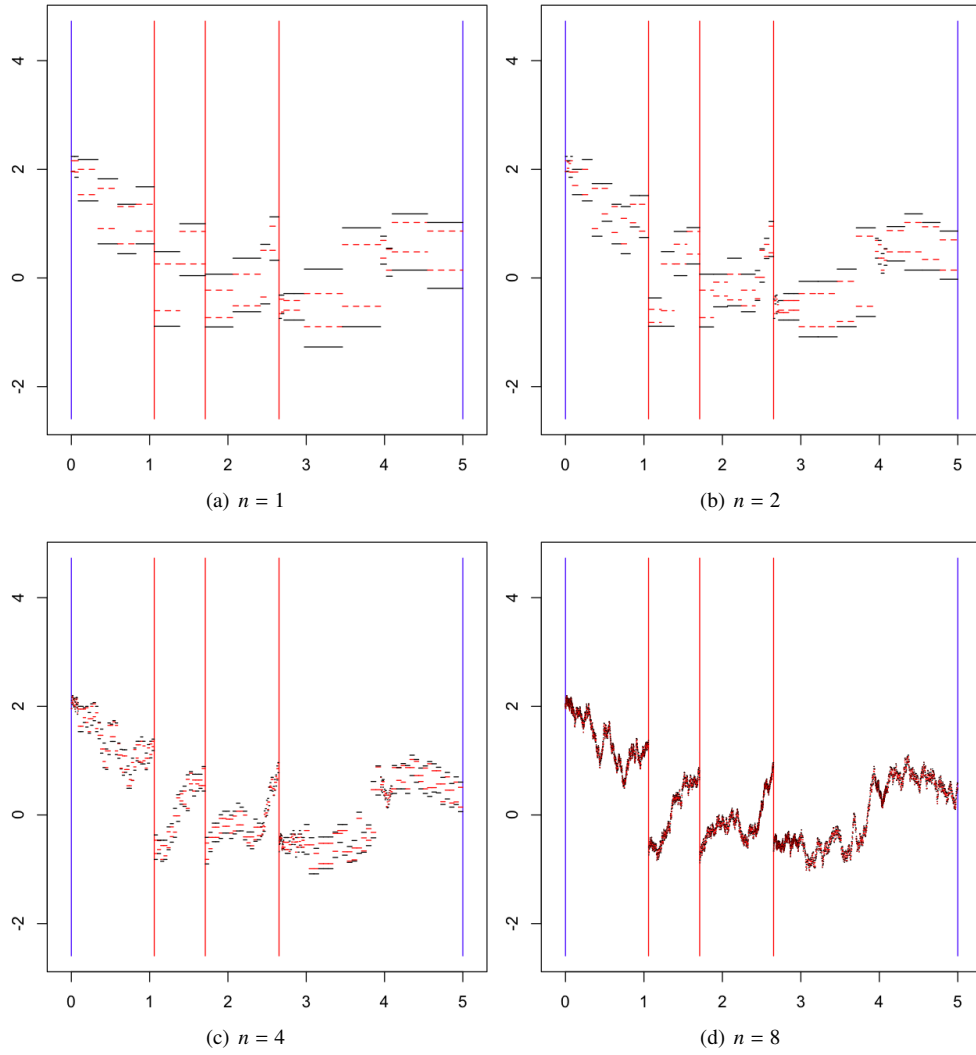


Figure 8.1.2: ϵ -strong simulation of jump diffusion sample path in Figure 8.1.1. n denotes the number of simulated bisections as per Algorithm 8.1.

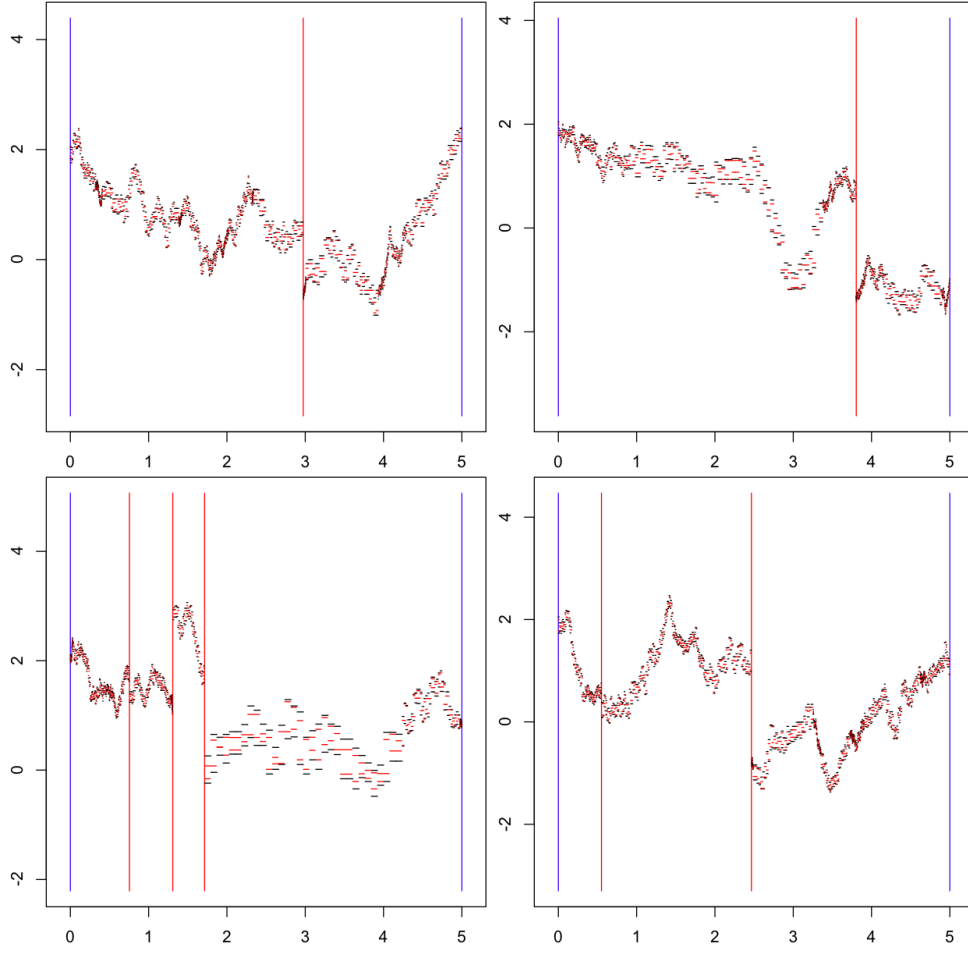
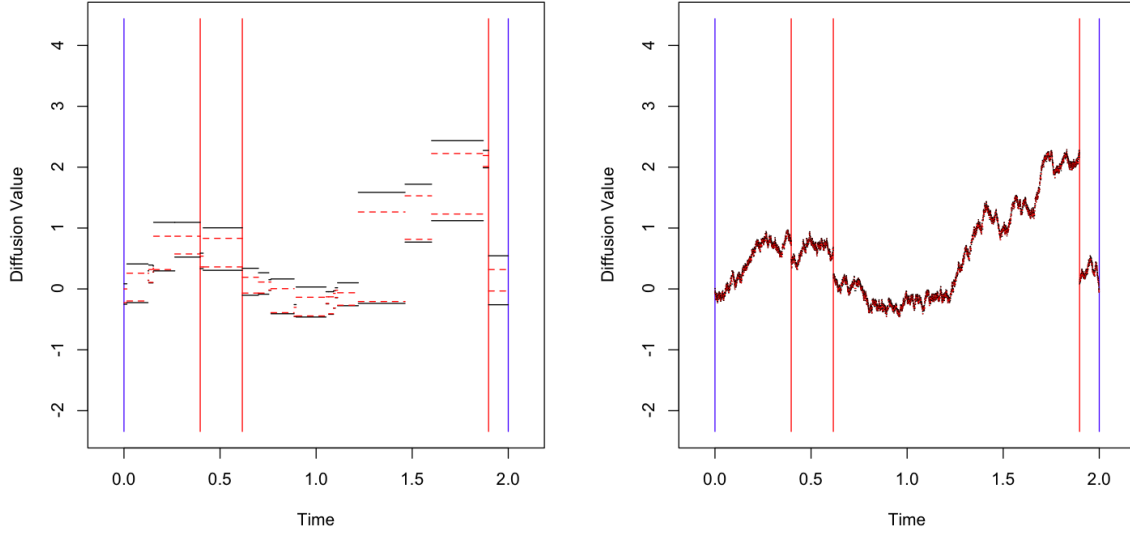


Figure 8.1.3: Sample path skeletons generated from the measure induced by (79) with $n = 6$ bisections



(a) Sample path skeleton simulated from the measure induced by (80).

(b) ϵ -strong simulation of (a) to tolerance $\epsilon = 0.05$

Figure 8.1.1: Vertical blue lines are the start and end point of the skeleton, vertical red lines are accepted jump times, horizontal black lines are almost sure bounds constraining the sample path and horizontal dotted red lines concern the layer attained by the sample path minimum and maximum in any given sub-interval.

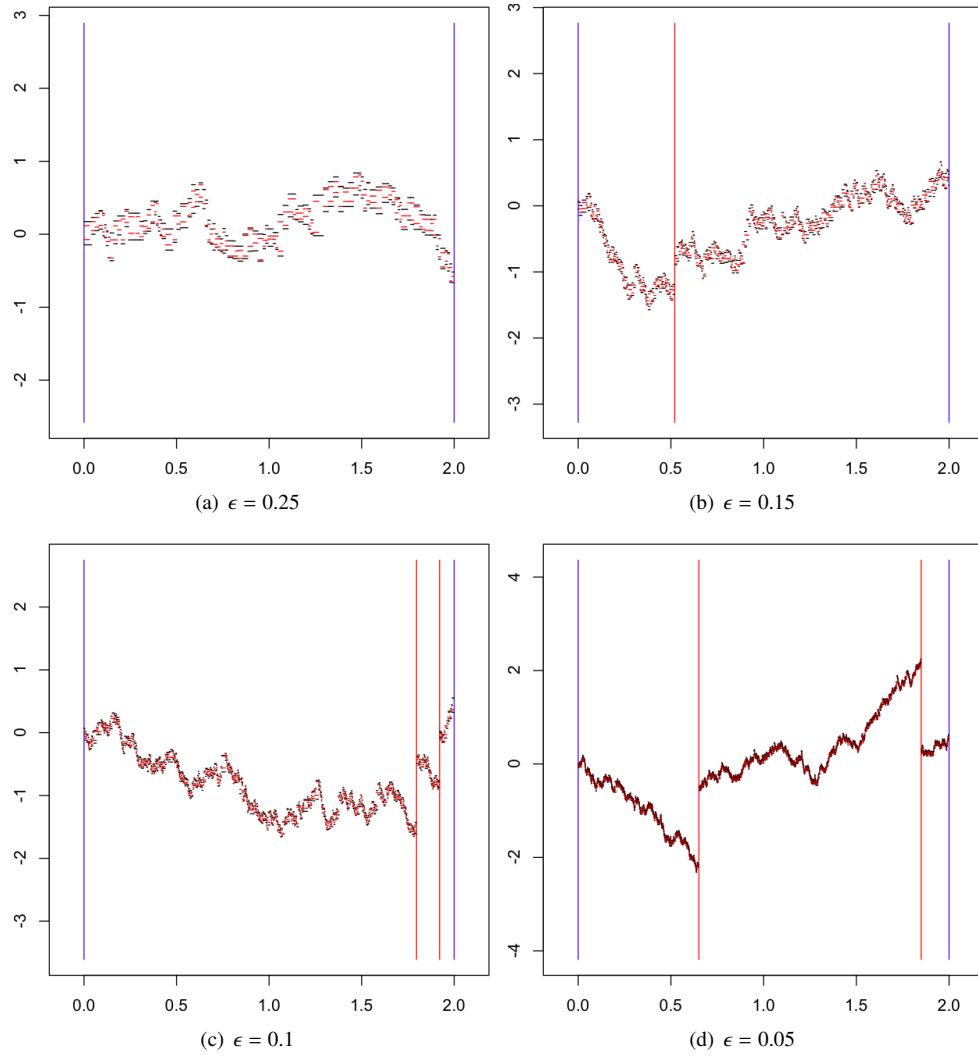


Figure 8.2.2: ϵ -strong simulation of jump diffusion sample paths from the measure induced by (80)

References

- [1] Y. Aït-Sahalia. Closed-form likelihood expansions for multivariate diffusions. *The Annals of Statistics*, 36:906–937, 2008.
- [2] T.W. Anderson. A modification of the sequential probability ratio test to reduce the sample size. *Annals of Mathematical Statistics*, 31(1):165–197, 1960.
- [3] S. Asmussen, P. Glynn, and J. Pitman. Discretization error in simulation of one-dimensional reflecting Brownian motion. *Annals of Applied Probability*, 5(4):875–896, 1995.
- [4] O.E. Barndorff-Nielsen and N. Shephard. Power and bi-power variation with stochastic volatility and jumps. *Journal of Financial Econometrics*, 2(1):1–37, 2004.
- [5] V. Beneš. Existence of optimal stochastic control laws. *SIAM Journal on Control*, 9(3):446–472, 1971.
- [6] A. Beskos, O. Papaspiliopoulos, and G.O. Roberts. Retrospective exact simulation of diffusion sample paths with applications. *Bernoulli*, 12:1077–1098, 2006.
- [7] A. Beskos, O. Papaspiliopoulos, and G.O. Roberts. A factorisation of diffusion measure and finite sample path constructions. *Methodology and Computing in Applied Probability*, 10:85–104, 2008.
- [8] A. Beskos, S. Peluchetti, and G.O. Roberts. ϵ -strong simulation of the Brownian path. *Bernoulli*, In Press.
- [9] A. Beskos and G.O. Roberts. An exact simulation of diffusions. *Annals of Applied Probability*, 15(4):2422–2444, 2005.
- [10] F. Black and M. Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3):637–654, 1973.
- [11] Z.A. Burq and O. Jones. Simulation of Brownian motion at first passage times. *Mathematics and Computers in Simulation*, 77:64–71, 2008.
- [12] B. Casella and G.O. Roberts. Exact simulation of jump-diffusion processes with Monte Carlo applications. *Methodology and Computing in Applied Probability*, 13(3):449–473, 2010.
- [13] N. Chen and Z. Huang. Localisation and exact simulation of Brownian motion driven stochastic differential equations. *Mathematics of Operational Research*, In Press.
- [14] L. Devroye. *Non-Uniform Random Variate Generation*. Springer, 1st edition, 1986.
- [15] B. Eraker, M. Johannes, and N. Polson. The impact of jumps in volatility and returns. *The Journal of Finance*, 58(3):1269–1300, 2003.
- [16] P. Eto and M. Martinez. Exact simulation of one-dimensional stochastic differential equations involving the local time at zero of the unknown process. *ArXiv e-prints 1102.2565*, 2011.
- [17] K. Giesecke and D. Smelov. Exact sampling of jump-diffusions. *Operations Research*, In Press.
- [18] A. Golightly and D.J. Wilkinson. Bayesian sequential inference for nonlinear multivariate diffusions. *Statistics and Computing*, 16(4):323–338, 2006.
- [19] A. Golightly and D.J. Wilkinson. Bayesian inference for nonlinear multivariate diffusion models observed with error. *Computational Statistics & Data Analysis*, 52(3):1674–1693, 2008.
- [20] F.B. Gonçalves and G.O. Roberts. Exact simulation problems for jump-diffusions. *Methodology and Computing in Applied Probability*, In Press.
- [21] M. Johannes, N. Polson, and J. Stroud. Optimal filtering of jump-diffusions: Extracting latent states from asset prices. *Review of Financial Studies*, 22(7):2259–2299, 2009.
- [22] I. Karatzas and S. Shreve. *Brownian Motion and Stochastic Calculus*. Springer-Verlag, New York, 2nd edition, 1991.

- [23] J.F.C. Kingman. *Poisson Processes*. Clarendon Press, 1st edition, 1992.
- [24] P.E. Kloeden and E. Platen. *Numerical Solution of Stochastic Differential Equations*. Springer, 4th edition, 1992.
- [25] R.C. Merton. Theory of rational option pricing. *Bell Journal of Economics and Management Science*, 4(1):141–183, 1973.
- [26] R.C. Merton. Option pricing when underlying stock returns are discontinuous. *Journal of Financial Economics*, 3(1):125–144, 1976.
- [27] B. Øksendal. *Stochastic Differential Equations*. Springer, 6th edition, 2007.
- [28] B. Øksendal and A. Sulem. *Applied Stochastic Control of Jump Diffusions*. Springer, 2nd edition, 2004.
- [29] U. Picchini, A. Gaetano, and S. Ditlevsen. Stochastic differential mixed-effects models. *Scandinavian Journal of Statistics*, 37(1):67–90, 2009.
- [30] E. Platen and N. Bruti-Liberati. *Numerical Solution of Stochastic Differential Equations with Jumps in Finance*. Springer, 1st edition, 2010.
- [31] K. Pötzelberger and L. Wang. Boundary crossing probability for Brownian motion. *Journal of Applied Probability*, 38:152–164, 2001.
- [32] B.D. Ripley. *Stochastic Simulation*. John Wiley, 1st edition, 1987.
- [33] C.P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, 2nd edition, 2004.
- [34] G. Sermaidis, O. Papaspiliopoulos, G.O. Roberts, and P. Fearnhead. Markov chain Monte Carlo for exact inference for diffusions. *Scandinavian Journal of Statistics*, In Press.